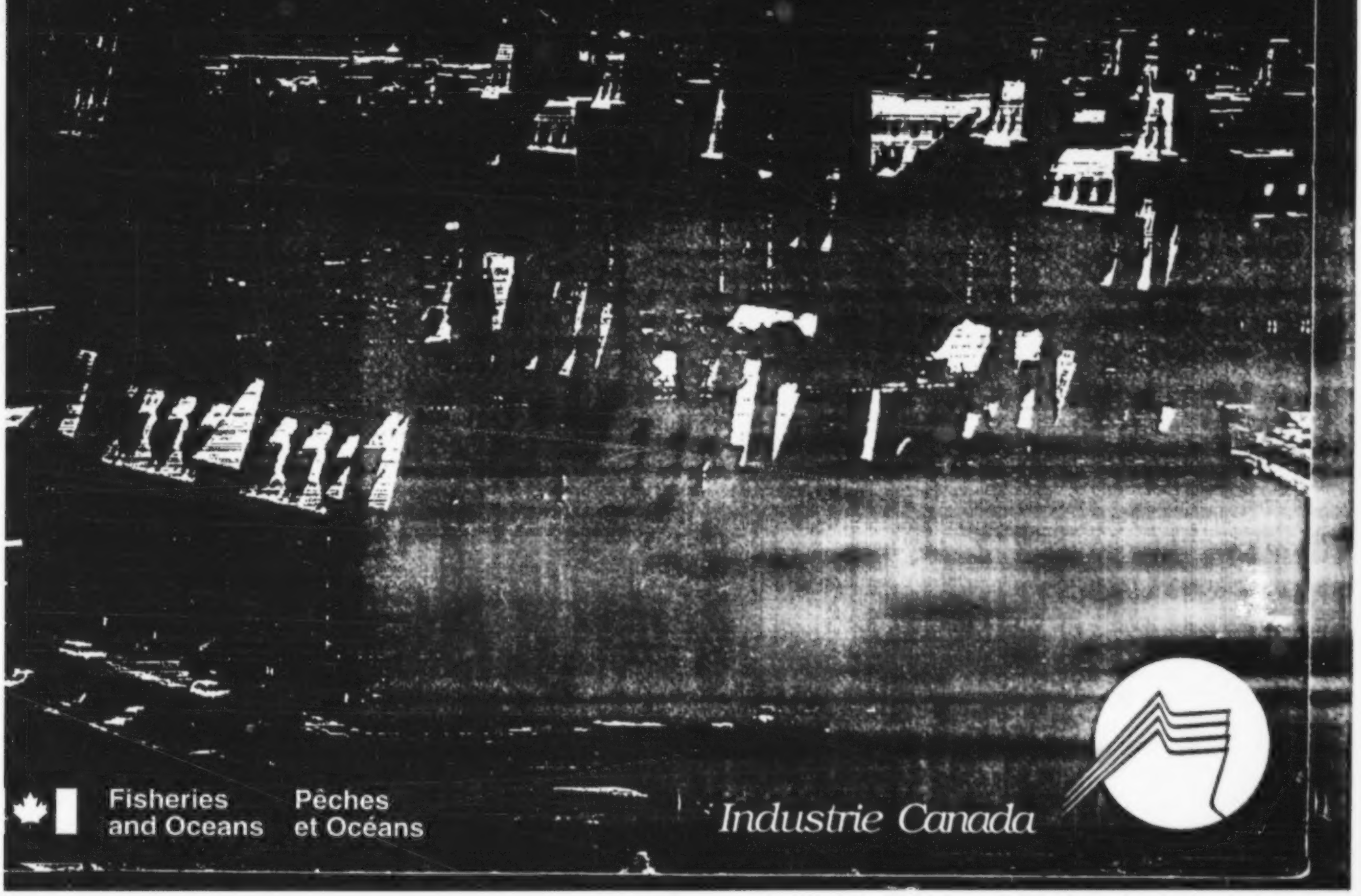
nowledge-Bas [illegible]
ystems & Robo [illegible]



Fisheries and Oceans
Pêches et Océans

Industrie Canada

311150

Proceedings of the Knowledge-Based Systems & Robotics Workshop

November 14 - 17, 1993

Edited by:

H. Merklinger
M. Farooq
P. Roberge
J.J. Grodski
Gen R. Dobson (ret)

Fisheries and Oceans Pêches et Océans

TRUTH · DUTY · VALOUR

Industrie Canada

FOREWORD

Research conducted to date on the international scene has clearly shown the growing presence of new information technologies in all areas of our lives. Advances in informatics and telecommunications have profoundly affected all economic sectors, whether they involve the manufacturing of goods, the provision of services or the acquisition and employment of defence resources. The changes are not limited to these areas; recent advances in Knowledge-Based Systems and Robotics highlight new opportunities for dual applications of these technologies in times of peace or conflict. There are already clear indications that we will alter our internal and external relations among government departments, industry and educational institutes.

With the integration of informatics and communications technologies, knowledge and information have become strategic resources and agents for change as our post-industrial society enters a new stage—the information age. The information society is "information intensive", which is to say that information will henceforth be the raw material for the development of new defence applications.

It is in this context that we in CRAD, with cooperation from Industry Canada, Fisheries and Oceans Canada and the Royal Military College of Canada, are pleased to bring together so many different agencies interested in developing and supporting our initiatives. Our primary goal is to ensure the availability and integration of advanced technologies to meet the entire spectrum of military needs. To accomplish this goal we are very conscious of the necessity to meet the operational and support requirements of the Canadian Forces economically in the present, while at the same time recognizing our responsibilities to advise our client of the very best and

AVANT-PROPOS

Les recherches effectuées sur la scène internationale ont clairement démontré que les nouvelles technologies de l'information influencent de plus en plus tous les aspects de notre vie. Les progrès de l'informatique et des télécommunications ont profondément transformé tous les secteurs économiques, qu'il s'agisse de la fabrication de biens, de la prestation de services ou de l'acquisition et de l'utilisation des ressources de défense.

Ces domaines ne sont pas les seuls à avoir changé; les progrès récents dans les domaines des systèmes à base de connaissances et de la robotique ont mis en lumière de nouvelles possibilités d'appliquer ces deux technologies de concert, en temps de paix comme en temps de conflit. Il est déjà manifeste que nous allons modifier notre façon de communiquer, tant dans les ministères que dans l'industrie et les établissements d'enseignement.

À la suite de l'intégration de l'informatique et des technologies des communications, la connaissance et l'information sont devenus des ressources stratégiques et des agents de changement, à mesure que notre société post-industrielle entre dans une nouvelle ère, l'âge de l'informatique. Cette nouvelle société repose sur l'information; cette dernière sera donc la matière brute à partir de laquelle seront mises au point les nouvelles applications en matière de défense.

C'est dans ce contexte que le CR Dév, en collaboration avec Industrie, Sciences et Technologie Canada, Pêches et Océans et le Collège militaire royal du Canada, a le plaisir de réunir autant d'organismes intéressés à réaliser et à soutenir ses projets. Notre principal objectif est d'intégrer les technologies avancées et de les mettre à la disposition des Forces canadiennes afin de satisfaire tout l'éventail de leurs besoins militaires. Nous sommes très conscients que most promising technologies that can be developed to meet these needs in future.

I appreciate all contributions from the private sector, other government departments and universities; we all benefited from your activities over this three-day workshop, and the associated exhibition. I am confident that your efforts helped to develop a synergistic approach to identifying practical applications of advanced technologies in Knowledge-Based Systems, Artificial Intelligence and Robotics.

pour atteindre ce but, nous devons dès à présent satisfaire à un coût raisonnable les besoins des Forces canadiennes en matière d'opérations et de soutien. De plus, nous savons que nous avons d'autres responsabilités; nous devons en effet tenir nos clients au courant des technologies les meilleures et les plus prometteuses qui pourront être mises au point afin de continuer à satisfaire les besoins précités.

J'apprécie toutes les contributions venant du secteur privé, des autres ministères et des universités; vous nous avez beaucoup apporté au cours de cet atelier de trois jours et de l'exposition qui l'accompagnait. Je suis convaincu que vos efforts ont contribué à établir une approche synergique qui permettra de trouver des applications pratiques pour les technologies avancées en matière de systèmes à base de connaissances, d'intelligence artificielle et de robotique.

K. A. Peebles

K. A. Peebles
Chief, Research and Development. Le chef - Recherche et développement.

ACKNOWLEDGEMENTS

The organizers of the Knowledge Based Systems and Robotics Workshop wish to express their gratitude to the Chief of Research and Development, Industry Canada, Fisheries and Oceans, and Foreign Affairs for sponsoring the Workshop.

The organizers are indebted to M. Farooq and P. Roberge of the Royal Military College, J. Grodski of the Defence and Civil Institute of Environmental Medicine and their Technical Committee Members for the organization, and review of responses to the Call For Papers and the subsequent design and conduct of the Academic Programme.

A special thanks to Key Note Speakers, Session Chairs, Panel Members, Authors, and to all who presented papers at the Workshop. We also acknowledge and appreciate the participation of many members of the Knowledge Based Systems and Robotic Communities representing government, industry, university and military researchers and program co-ordinators.

We are also grateful to BGen (Retd.) R. Dobson of B.I.S.I. and to S. Akhtar from Industry Canada for facilitating the display and promotion of dual use advanced technologies and related hardware. Our thanks is extended to Marg Coll of Conference Coll Inc. for processing advanced and on-site registration.

Whitin the CRAD organization we are indebted to many who devoted a lot of time and effort to guarantee a successful Workshop. Sub-committees were supervised by R. Elliot, D. Wakefield, T. Davies and P. Dumouchel. Secretarial support was ably provided by D. Hill.

H. Merklinger
Workshop Chair

REMERCIEMENTS

Les organisateurs de l'Atelier sur les systèmes des bases de connaissance et la robotique expriment leur plus profonde reconnaissance au Chef-Recherche et Développement, aux Ministères de l'Industrie du Canada, des Pêcheries et Océans et des Affaires Extérieures pour leur support et subventions.

Les organisateurs sont redevables à M. Farooq et P. Roberge du Collège Militaire Royal, J. Grodski de l'Institut de Défense et Civile de Médecine Environnementale ainsi qu'aux membres de leur comité technique pour l'organisation et la revision des soumissions, et la conception subséquente du programme académique ainsi que sa conduite.

On remercie particulièrement les conférenciers invités, les présidents de session, les membres de jury et tous ceux qui ont présenté des articles durant l'atelier. Nous sommes reconnaissants de la participation de plusieurs membres de la communauté de systèmes des bases de connaissance et robotique représentant le gouvernement, l'industrie, les universités et chercheurs pour la défense militaire.

On est de plus reconnaissant au Bgén (retraité) R. Dobson de B.I.S.I. et à S. Akhtar du Ministère de l'Industrie du Canada d'avoir permis l'exposition et la promotion des technologies avancées à utilité dualle et le matériel qui s'y rapporte. Nos remerciements vont aussi à Marg Coll de Conférence Coll Inc. pour l'inscription anticipée et l'inscription locale.

Parmi l'organisation de CR Dév nous sommes reconnaissants à ceux qui ont donné le temps et les efforts nécessaires à la garantie du succès de l'atelier. Les sous-comités étaient supervisés par R. Elliot, D. Wakefield, T. Davies et P. Dumouchel. Le secrétariat fut supporté par D. Hill.

H. Merklinger
Président de l'Atelier

CONTENTS

Foreward K.A. Peebles, CRAD

Acknowledgements H. Merklinger

Plenary Session I Chair: George Kimball (Assoc CRAD)

The Role of Knowledge-Based Systems and Robotics in Future Military Systems 1
Keynote Speaker - MGen R.N. Fischer, Assoc ADM (Mat)

Session I - Expert Systems I **Chair:** R. Hayashi, DREA

An Expert Network Analyzer: Knowledge Acquisition, Fault Diagnosis and Prediction 5
N.J. Dimopoulos, K.F. Li, S. Neville, A. Watkins, Z. Lu, D. Yamakawa, U Vic

Distribution Planning: A Constrained Heuristic Search Approach 13
M.S. Fox, U of T, V. Saks, I. Johnson, Carnegie Group, Pittsburgh PA

A Knowledge Based System for Military Aircraft Load Planning 21
Capt. J. Richardson, M. Labbé

Potential Applications of Knowledge Based Systems for the Performance Evaluation of Materials in Naval Systems 29
K.R. Trethewey, Royal Naval Engineering College, Plymouth U.K.; P.R. Roberge, RMC

Session II - Sensors I **Chair:** D. Graham, Thomson-CSF

Range Sensor Array for an Experimental Mars Rover 39
L. Korba, NRC; W. McMath, S. Yeung, CSA

Development of a Vision Guided Target Detector and Tracker 47
R. Ball, L. Pelkowitz, IMAGO Machine Vision Research; S. Penzes, DRES

On-Line Robot Trajectory Generation Through Real-Time Optical Tracking 55
N. Krouglicof, A. Desrochers, D. Richard, U de Québec

Tactile and Proximity Sensor System 63
N. Paquin, J. Blezius and D. Sfeir, MPB Technologies Inc; P. Wojcik, K. Chrystall, AB Research Council

Session III - Neural Networks I **Chair:** G. Akhras, RMC

Track Maintenance Using Real-Time Neural Computing 71
J. Liu, B.A. Bowen, CompEngServ Ltd

A Neural Network Approach to Detect Traffic Anomalies in a Communication Network 77
Capt F. Viens, CMR; B. Pagurek, N. Dawes, Carleton U

A Neural Network Used to Model the Release of Fission Products from Nuclear Reactor Fuel Under Severe Accident Conditions 85
W.S. Andrews, B.J. Lewis, RMC; D.S. Cox, AECL

Autonomous Impedence Control of a Robot in an Unstructured Environment Using a Qualitative Task Model 93
S.P. Ananthanarayanan, A. A. Goldenberg, J. Mylopoulos, U of T

Session IV - Control I Chair: C. J. Damaren, RRMC

Reinforcement Learning and the Parametric Motor Control Hypothesis Applied to Robotic Arm Movements 101
J. Lipitkas, G.M.T. D'Eleuterio, U of T; O. Bock, York U; J.J. Grodski, DCIEM

Integrated Mechanical and Control Design for Well-Designed Flexible Joint Robots 107
H.A. ElMaraghy, F. Liang, McMaster U

Remote Mine-Hunting Systems - Vehicle Stability Trials 115
J.M. Preston, DREP; L. Shupe, International Submarine Engineering Research Ltd

Both Planning and Supervised Autonomous Control of a Robotic Arm in Space 123
H.S. Gaskill, MUN; C.P. Dawson, Instrumar Ltd

Speaker - Knowledge/Technology Infrastructure: Perspectives on the Past and Future

J.E. Smith, NRC 131

Session V - AI & Real-Time Applications Chair: L. Lamontagne, DREV

Learning Machines Based on the Evolving Transformation Model 137
V.N. Kamat, V.C. Bhavsar, L. Goldfarb, UNB

General Principles in the Design of Real-Time Knowledge-Based Systems for the Threat Evaluation and Weapon Assignment Process 145
R. Carling, DREV

Les Stratégies de résolution de conflits utilisées avec une restriction de type temporel 153
M. Bélanger, CRDV

Real-Time Expert System Shells: Evaluation Criteria and Results for Commercially Available Tools 159
R.G. Brown, Software Kinetics Ltd; M. Bélanger, DREV

Session VI - Mobile Robots Chair: D. MacKay, DRES

Subsea Robotics Research at OERC/MUN 167
M. Hinchley, MUN

Conception d'un mini-véhicule pour l'étude de systèmes de guidage autonome 173
R. Hurteau, P. Cohen, S. Gill, École Polytechnique; J.-C. Piedboeuf, RMC

ANCAEUS Navigation Module 179
T.J. Nelson, APRO Applied Robotics Inc

Dead Reckoning for Mobile Robots via Sensor Fusion 183
D.N. Green, J.Z. Sasiadek, Carleton U; D.S. Necsulescu, B. Kim, U of Ottawa

Session VII - Power Sources Chair: M. B. Broughton, RMC

Electrochemical Power Sources for Robotic Systems 189
A. Anantaraman, W.A. Adams, U of Ottawa

Power System Health Monitoring and Optimization Using Artificial Neural Network 197
R. Cunningham, Pernix Technologies Ltd; R. D. Mason, GenLink Ent Ltd

Powering Autonomous Underwater Vehicles with Aluminum 205
R.G. Laidley, NDHQ; J.P. Salvador, P.R. Roberge, RMC

Simulation Based Artificial Intelligence for Intelligent Power System Projection 211
J. Hensler, Hensmand Ltd; LCdr M. Tinney, NDHQ

Session VIII - Artificial Intelligence I Chair: S. Abu-Hakima, NRC

An Inductive Learning Approach to SARSAT Confidence Factors 219
I. Taylor, ATGHQ; H. Zhao, ABIT System Inc

A Plan Evaluation Methodology for Military Command & Control 225
Maj R. Weisman, NDHQ; J. Fugere, RMC

Decision Aids for Intelligence Fusion 237
M. Gauvin, D. Roy, DREV; M. Lizotte, DMR Group Inc

The Case-Based Planning System: Using Case-Based Reasoning and Genetic Algorithms to Plan Spacecraft Operations 245
P. Adamovits, CSA; R. Jetzelsperger, Software Kinetics Ltd

Session IX - Telerobotics/Navigation Chair: A. Poirier, CSA

A Potential Supervised Autonomous Robotic System for the MSS 253
D.P.W. Graham, S. Seifu, Thomson CSF Systems; N. Burtnyk, NRC

Virtual Telerobotic Control 261
A. Rastogi, P. Milgram, D. Drascic, U of T; J. J. Grodski, DCIEM

Defence Teleoperation and Stereoscopic Video 269
J.J. Grodski, DCIEM; D. Drascic, Human Engineering Research & Consulting

Operator-Robot Interaction for Task Level Telerobotics 277
P. Freedman, Centre de recherche informatique de Montréal

Project JINGOSS: A Tele-Operated Metallic Mine Detector (U) 283
A.Carruthers, R. Eirich, Y. Das, DRE Suffield; A. Kramer, Oliphant Management Services

Session X - KBS in Robotics Chair: K. Bryden, DREO

Paradigms of Intelligent Robotics 293
J.S. Basran, E. Petriu, U of Ottawa

Système à base de connaissances pour la planification des taches avec applications en robotique 301
M. C. Murgulescu, Ecole Polytechnique

Functional Abstraction for High-Level Online Robot Programming 309
C. Archibald, NRC; E. Petriu, U of Ottawa

Knowledge Based Task Planning for Supervisory Control of Telerobotic Systems 315
P. Feighan, K. Chrystall, A. Dagnino, AB Research Council

Design Knowledge Capture of Robotic Systems 323
E. Yaniv, W.R. Cyre, Virginia Tech

Session XI - Fault Diagnosis Chair: P. Adamovits, CSA

Cracks, Weld Defects and Porosity Classifiers for Submarine Pressure Hull Steels 329
J. Helie, D.R. Hay, Tektrend International; G. Labbé, J.F. Boucher, NDHQ; J.L. Kennedy, J.R. Matthews, DREA

Multiple Views of Knowledge in Diagnosis 337
S. Abu-Hakima, NRC

OMNIEX: An Expert System for Automated Fault Prediction and Diagnosis 345
J. Boily, B. Graham and B. Marrocco, Spectrum Engineering Corp Ltd

Expert System for Digital Circuit Troubleshooting in an Interactive Learning Environment 351
K. Sandrasegaran, A. Malowany, McGill U

A Simulation-Based Approach to Space Craft Power System Fault Detection and Diagnosis 359
P. Adamovits, CSA

Designing Knowledge-Based Advisory Systems for Diagnostic Problems in Engineering 367
G. Akhras, H.C. Foo, RMC

Plenary Session II Chair: G. Akhras, RMC

Hybrid Artificial Intelligence & Military Applications 375
J.E. Bowen, B.A. Bowen, J. Liu, CompEngServ Ltd

Session XII - Sensors II Chair: S. Elgazzar, NRC

Development of a Vision Guided Robotic Assembly Workcell 383
R. Ball, T. Folinsbee, IMAGO Machine Vision Research Inc; T. Pryor, Sensor Adaptive Machines Inc

Développement d'un capteur d'efforts intégré pour le contrôle adaptatif de trajectoires robotisées 389
A. Desrochers, N. Krouglicof, D. Croteau, U de Québec

Estimation of EndPoint Position and Orientation of a Flexible Link Using Strain Gauges 397
J.-C Piedboeuf, S. Miller, RMC

Stochastic Position Sensing in Unknown Environments 405
T.J. Nelson, APRO Applied Robotics Inc

Model-Based Visual Recognition of 3-D Objects Using Pseudo-Random Grid Encoding 411
N. Trif, E.M. Petriu, U of Ottawa; W.S. McMath, S.K. Yeung, CSA

Session XIII - Overview in Robotics Chair: Col M.K. Jeffery, DLR

Projets Belges MDN de dévelopment en robotique intérieure et extérieure 419
Y. Baudoin, Ecole Royale Militaire de Brussels

NATO Research Study Group on Operator-Robot Interaction 429
J.J. Grodski, DCIEM

US-Canada Cooperation Opportunities in Defence Robotics 435
R.M. Phelps, US Army; J.J. Grodski DCIEM

U.S. Unmanned Ground Systems - A Strategy for System Development and Fielding 441
LtC M.L. Swinson, US Army

Defence Robotics in Canada: An Overview 447
J.J. Grodski, DCIEM

Etude PROTECTION: Définition de Plates-Formes Robotisées Télé-Opérées Bas Coût pour le Champ de Bataille 461
A. Carrel, J.P. Pouplard, ITMI; J.L. de Moulins, A. Lelarge GIAT Industries;
G. Falcou, M. Dégé, ECA (France)

Session XIV-Expert Systems & Info Management Chair: M. Laplante, G.E. Harris,
RES Policy Research Inc

An Analysis Tool for the Backbone of the Automated Defence Data Network 471
K. Bryden, CRC; D. Wattie, DND

A Distributed Coordination Model for Battle Staff Tasks 477
L. Lamontagne, DREV

Aurora: A Multi-Agent Prototype Modelling Crew Interpersonal Communication Network 485
D. Gagné, G. Nault, A. Garant, J. Desbiens, CMR

An Expert System for Condition Monitoring 493
C.G. Selkirk, P.R. Roberge and G.F. Fisher, RMC; S.K. Yeung, DREP

EXCLASS - Knowledge Based Job Classification and Evaluation 501
G.E. Harris and M. Laplante, RES Policy Research Inc

Concept-Based Composition of Job Descriptions With Automatic Text Generation 509
T. Korelsky, D.E. Caldwell, CoGentex Inc

Lessons Learned Using an Expert System Shell 515
J.R. Amyot, W. Boev, NRC

Panel on Knowledge Based Systems Chair: E. Solem (DSEA)

Knowledge-Based Systems for Managing Complexity 523
E. Solem, DSEA

Expert Systems in Developing Countries: Learning Through the IDRC's Experience 529
Z. Mikolajuk, IDRC

Learning Organizations and the Management of Knowledge 537
G. Thomas, RES Policy Research Inc

Panel on Robotics Chair: M. Farooq, RMC

Robotics: The Present and the Future 541
M. Farooq, RMC

Session XV - Modelling & Simulation Chair: K. Yeung, DREP

Simulation of Flexible-Link Manipulators with Inertial and Geometric Nonlinearities 547
C. Damaren, RRMC; I. Sharf, U Vic

A Design Methodology for the Development of Experimental Robotic Systems 555
S.J. Miller, J.-C. Piedboeuf, RMC

A Practical Dynamic Model for the Development of Experimental Robotic Systems 565
G. Hardy, DREP

Computational Aspects of Real-Time Robot Dynamics Simulation 573
X. Cyril, M. Huculak, R. Megley, CAE Electronics

Robot Interface for CAD/CAM Integration 581
M. Ferland, Hydro-Québec and J. O'Shea & M. Rabemanantsoa, Ecole Polytechnique

Session XVI - Expert Systems II Chair: M. Labbé, CMR

Application of Object Oriented Analysis and Design Methodology to Knowledge Based Systems Development 587
The Cable Run Intelligent Advisor System
S. Jones, C. Bassias, R. Sassine, C. Tousignant, Paramax Systems Inc

CLAP: Une implantation du modèle acteur sur réseau hétèrogène 593
J. Desbiens, M. Toulouse, D. Gagné

Knowledge Representation Issues in Blackboards for ESM 601
G. Bourassa, Lockheed Canada

Knowledge Acquisition and Conceptual Design for a TACAN Siting Advisor System 611
W.H. Henneker, C.S. Phan, NRC

An Object-Oriented Knowledge Based System for the Evaluation of the SCC Intensity of High Strength Aluminum Alloys 617
P.R. Roberge, RMC

Vers une architecture d'Agent pour le commandement et contrôle (C^3) 625
E. Paquet, P. Levesque, B. Chaib-draa, U de Laval; L. Lamontagne, CDRV

Session XVII - Mobile Robots/Obstacle Avoidance Chair: K. Foster, CRAD

Floor Obstacle Avoidance for a Mobile Robot 633
A. Manz, D.A. Green, S. Elgazzar, F. Blais, M. Lecavalier, NRC

C-Space Obstacle Computation: A Geometry-Based Fast Algorithm 641
C.S. Zhao, M.M. Bayoumi, Queen's U; M. Farooq, RMC

Development of an Enhanced Mobility Testbed 649
C, Brosinsky, A.W. McCormac, D.M. Hanna, DRES; D.R. Fenrick, HF Automotive Performance Research Inc

Plenary Session III Chair: J. Grodski, NRC

Beyond the Space Age... Environmental Robotics: Nuclear Site Remediation 655
Keynote Speaker - M. Brooks, NRC

Session XVIII - Artificial Intelligence II Chair: P. Adamovits, CSA

An Architechture for Adding Intelligence in a Multiprocessor Envronment 657
E. Suen, CalCorp, R. Hayashi and A. Dent, DREA

A Method of Interactive Recognition of Spatially Defined Model Deployment Templates Using Abstraction 665
S. Woods, DREV

Toward Time-Dependent Planning of Missile Allocation and Engagement Scheduling in a Naval Threat Evaluation and Weapon Assignment System 677
B.A. Chalmers, DREV

Artificial Intelligence Applied to Sonar Data Management 685
M. Laplante, G.E. Harris, RES Policy Research Inc

Session XIX - Control II/Obstacle Avoidance Chair: J.-C. Piedboeuf, RMC

Kinematic Analysis and Controller Synthesis for Path Tracking in Automated Vehicles with Double Steering 693
A. Hemami, Ecole Polytechnique de Montréal

Basic ROMAC Control Algorithms 701
B. Hall, B Hall Industries; J.J. Grodski, DCIEM

Trajectoires optimales de robot en utilisant le concept d'isotropie 709
J. Poulin, M. Lortie, P. Gely, Ecole de Technologie Supérieure

Fuzzy Logic for Particle Accelerator Control 715
P.C. Lind, W.F.S. Poehlman, McMaster U

Session XX - Neural Networks II Chair: A.L. Jenkins, RMC

Finding Neural Network Weights Without Training 723
L. Jenkins, Maj S. Jones, RMC

Face Identification Based on an Interactional Neural Network 731
P. Boulanger, NRC

Neural Network and Knowledge-Based Systems for Signal Validation in the Nuclear Industry 739
H. Dai, J.W. Thompson, Atlantic Nuclear Services Ltd

Control of Multi-DOF Robots Using Neural Networks 747
K. Kiguchi and D. Necsulescu, U of Ottawa

Closing Session Chair: M. Farooq, RMC &
H. Merklinger, CRAD

Workshop Summary - Col Mialkowski 755

900C57

The Role of Knowledge-Based Systems and Robotics in Future Military Systems

MGen R. N. Fischer
Associate Assistant Deputy Minister Materiel
National Defence Headquarters, MGen G. R. Pearkes Building.
Ottawa, Ontario, K1A 0K2.

#149C07

Abstract

Our world, and the battlefield, have changed drastically in recent years. The context of our time places high value on security with minimal cost to the taxpayer. Knowledge-Based Systems and Robotics, both separately and jointly, hold significant promise to help meet these requirements. This paper offers one man's views of the place of these technologies in future military systems.

Introduction

Thank you Ken for your kind introduction.

Welcome to all participants, especially those from abroad. I had been invited to speak at the robotics workshop two years ago in Medicine Hat. I was prevented from attending that event and so I welcome this opportunity to share a few thoughts about "The Role of Knowledge-Based Systems and Robotics in Future Military Systems"

Context

Where do these technologies fit in the future armed forces of Canada and elsewhere? To answer such questions it is helpful to provide a context for possible future armed conflicts.

The incredible geo-political, social and technological changes we have witnessed during the past few years are on a scale which no one could have imagined only a decade ago. In spite of the fact that the weapons of war are more accurate, powerful and more terrifying than was the case in Vietnam, let alone World War II, we are expected to preserve the lives of our soldiers, sailors and airmen. We place a higher value on the safety and security of the individual than we have ever done in the past. Today's battlefield is a lethal environment in which the men and women, whose lives we value dearly, are asked to place their trust in the weapons and equipment we provide them.

In order to accomplish the military mission we have limited choices. We can defend our men and women through greater protection against increasingly capable weapons. We can destroy the weapons aimed against us before they reach their target. Or we can take our personnel out of harm's way altogether, and try to fight some form of remote controlled battle. In all likelihood, we will need to do a bit of all three. It would seem unlikely that we can take the human completely out of harm's way, therefore we must provide some form of protection. Through quick reaction and counter action, we can probably neutralize some portion of the weapons arrayed against us. And in specific situations there is a good likelihood that we can remove the 'operator' from the scene of conflict altogether.

For the past forty-five years—during the years 1947 to perhaps 1991—the world has lived under the threat of an imminent major world conflict. Two societies have lived in great fear and mistrust of each other. These fears resulted in massive investments in highly sophisticated and expensive weapon systems. The danger has, at least for the foreseeable future, diminished significantly. And it is the conviction of the citizens of many countries that we can now afford to relax our defences and enjoy the so-called peace dividend. And given the current economic situation, many governments are being asked to reduce defence budgets to protect social programs or reduce deficits.

Yet, as we see in the news media every day, the chances of our forces becoming involved in armed conflict seem to have increased rather than decreased.

As I see it, the consequences of all this are as follows:

- To preserve our values we must continue to develop our military capabilities.
- Armed conflicts are likely to be of a lesser intensity than that for which we were preparing even a few years ago.
- Our government, our people, will expect us to reduce defence-related spending, yet still expect us to fight and not get ourselves killed.

The message is clear: we will have to do it smarter and we will have to do it cheaper. We will be expecting knowledge-based systems to help us do it smarter. And we will be expecting robotics to remove the individual soldier from danger and to reduce the numbers required to carry out the mission.

During the cold war, we had to have technological superiority at almost any cost. In the post cold-war era, we have to maintain technological superiority at minimal cost.

The Technology Environment

You have all heard that in many fields, commercial technology is overtaking military technology. This is generally good news. It means that there will be many cases where we in the military can adopt commercial products and practices, thereby taking advantage of the economy of scale associated with the commercial marketplace. So-called "dual-use" technologies will have a very important place. We must be mindful, however, that there will be uniquely military applications for which there are no available commercial solutions. The international Research and Development community will have to identify those technological gaps and ensure that they are cost-effectively filled.

A Lesson from the past

I am told that the first "automatic camera" was one that opened the light path automatically at the mere squeezing of a rubber bulb, and closed it off again after a pre-determined interval of time. This automation saved the photographer from having to remove the lens cap manually, count out the seconds on his fingers, and then replace the lens cap over the lens. The automatic camera of today might well recommend the focal length of the lens to be used, set the lens to that value, focus where the photographer is looking at that instant, assess the light conditions against a database of perhaps ten-thousand prize-winning photographs, recommend the shutter speed and aperture to be used, decide if flash assistance is appropriate, amend settings if so requested, and then take three rapid-fire pictures using slight variations on the main recommendation, and finally confirm that it has completed the task without evident error.

One lesson here is that our expectations of automation increase with time. The automatic camera of today does things in an instant that were once not even imaginable.

Another lesson here is that the automatic camera is still under the control of its operator—for all its sophistication, it is powerless to act alone. Don't expect too much from your machine! Available technology has a lot to offer, but it must in general be matched to certain well-defined tasks.

Don't waste time and effort on the wrong problems.

Knowledge-Based Systems

To quote Gen. Colin Powell[1]:

> *The sight of a soldier going to war with a rifle in one hand and a laptop computer in the other would have been shocking only a few years ago. Yet that is exactly what was seen in the sands of Saudi Arabia in 1990 and 1991.*

The image is an important one. Yes, weapons are still part of warfare. But access to up-to-date, accurate, reliable information and the ability to manipulate it acts as a huge force-multiplier. To quote Gen. Powell again:

> *The ultimate goal is simple: Give the battlefield commander access to all the information needed to win the war. Give it to him when he wants it, where he wants it and how he wants it.*

Some military experts foresee huge linked information networks gathering all manner of information and updating it on a continual basis. The networks will offer:

- Intelligence about enemy status and activities,
- Own readiness and status (Who's where? What does he have? Is he ready to fight? etc.)
- Supply situation (Where are the supplies needed to sustain a battle? Do we have everything we need? How long will it take to get what we need etc.?)
- Weather,
- Ground conditions,
- A model of anticipated progress,
- A statement of the known unknowns,
- Standard tactics and plans,
- What if? A plan for contingencies, and
- Plans of related units (Can we expect support from X if we need it?).

The network would, of course, deny such information to the adversary, even if he should capture a node in the network!

It is one thing to have this information and another to manage it. The management of the information requires a network of knowledge-based systems to receive the information, interpret it, route it, combine it with existing and other new information, and distribute it in an intelligent fashion without overloading available communication paths. The information will need to be distilled down to just the essentials without leaving out anything important.

The system needs to be able to adapt to communication conditions. If only one low data rate path is available, the system must determine which information must be transmitted first. And the system needs to be able to respond to its clients' needs. The receiver of the information should be able to alter the priorities in a dynamic fashion. And the system needs to be robust; it must function even if it is not all there, and it must reject false information.

In such a Command-Control-Information system, formal knowledge-based systems would be at work at a number of levels:

- Sensor data interpretation,
- Fusion of sensor data,
- Extraction of military information from sensor data,
- Routing of data to appropriate destinations,
- Setting communication priorities based upon current situation and command status,
- Assembling and editing reports "from the front",
- Monitoring logistics,
- Generating stores lists, plans and orders, and
- Planning and scheduling Communications.

This is a tall order, but I believe it is achievable, incrementally and as a result of international and commercial cooperative efforts.

Other advantages of reliance on Knowledge-Based Systems include:

- Rapid transfer of knowledge and skills from one location to another,
- Uniform application of doctrine and standards,
- Preservation of expertise in spite of personnel turnover, and
- Standardization of training.

In the training environment the application of knowledge-based systems should bring a reduction in the number of training instructors necessary, and yet broaden the base of expertise on call. I expect knowledge-based systems to improve the quality of human decisions through relieving the operator from stress and overload by providing timely and tireless assistance. I also expect knowledge-based systems to help reduce mutual interference through the incorporation of IFF data in fire-control systems, as well as through immediate knowledge of the disposition of friendly forces.

Many of the general applications of knowledge engineering are being and will be applied in the commercial world. We will need to adapt what we can where we can, but remain mindful of the military requirement.

Robotics

On the robotics front, the following defence applications have been identified for some time:

- Relieving the human from routine repetitive and perhaps boring tasks,
- Reducing crew size,
- Multiplying the capabilities and effectiveness of a single individual,
- Carrying out tasks impossible for the human, and
- Carrying out tasks under conditions too life-threatening for the human.

Tasks for specialized tele-operated and autonomous robotic systems include:

- Mine laying, detection, and clearing,
- Placement of demolition charges,
- Vehicle loading,
- Vehicle guidance (robot convoy),
- Deception, decoys and spoofing,
- Automated surveillance,
- Unmanned artillery and artillery spotting,
- Fast-reaction defences,
- Bridging and obstacle clearance under fire,
- Unmanned ferries,
- Trenching,
- Cable-laying and repair,
- Road and runway repairs,
- General repair and maintenance of weapons systems and vehicles,
- Automated inspect/repair/inspect/test facilities,
- Decontamination, and
- Operations in chemical and radiation environments.

In many of these tasks technical success will mean reliability and low cost. Conducting any task under fire calls for redundancy, tolerance to minor damage, and relatively low replacement cost.

The adoption of civilian robotic machines may be limited. There is little question that commercial robots will operate in structured support environments, but their functioning under highly unstructured operational conditions—such as the modern battlefield—will undoubtedly call for a modification here or there. There is some hope that industrial robots will break out of the highly predictable and stable environment of the factory and find employment in less well-ordered environments such as agriculture and commercial mining. At the same time we would be well advised not to ignore the military significance of purely commercial robotics in the commercial environment. The ability to alter, increase and sustain production without increasing the workforce will help to support any extended conflict and should decrease the cost of military production.

Robotics Combined with Knowledge-Based Systems

Finally we come to the rationale for holding this combined workshop on Knowledge-Based Systems and Robotics. Clearly the robots will benefit greatly from embedded knowledge-based systems for guidance and control.

Knowledge-based systems provide the vision, purpose and control needed in the robot. Even the tele-operated robot needs to be a problem-solver. Knowledge-based systems offer the prospect of greater autonomy to resolve the unexpected, to accomplish the task under difficult and dynamic conditions. Knowledge-based systems hold the prospect of providing robots with an awareness of their surroundings and their progress.

Knowledge-based systems and robotics technologies are both characteristically multi-disciplinary. The broader the background and experience we can harness, the greater should be our progress in these areas.

Closing

In summary, there clearly are almost limitless applications for intelligent systems and robotics; the combination of the two in future military systems promises to revolutionize warfare. These technologies are of strategic importance; it is imperative that we understand both their benefits and their limitations. There are many scientific and engineering challenges ahead of us. We must maximize the extent to which we can build on commercial systems while remaining mindful that we will on occasion have requirements which extend beyond the commercial domain.

Canada will not "go it alone". International exchanges with our allies and close cooperation with industry are essential to making progress and keeping R&D costs affordable.

I wish you success in this workshop. Our expectations are high and we are counting on you to help provide cost effective knowledge-based robotic systems for the men and women of tomorrow's Armed Forces.

References

1. Powell, Colin L., "Information-Age Warriors" Byte magazine, p. 370, July 1992.

904206

An Expert Network Analyzer: Knowledge Acquisition, Fault Diagnosis and Prediction

Nikitas J. Dimopoulos, Andrew Watkins, Stephen Neville and Kin F. Li
Department of Electrical and Computer Engineering
University of Victoria
Victoria, B.C.

#149608

Abstract

In this work, we present the framework of an expert system that will be capable of diagnosing and predicting faults in a cable television plant.

The structure of the diagnostic expert system under development comprises equipment specific and network specific parts.

For the equipment specific part, there exist well established diagnosis and calibration procedures. These are captured and integrated into an equipment specific knowledge based diagnosis system with the aid of specifically designed knowledge acquisition system.

The network itself is monitored in real time, and its status data are used to determine failure points and the likely cause of the failure.

Introduction

Engineering systems are becoming numerous and complex. This has resulted in the proliferation of the number and complexity of procedures necessary for maintenance, diagnosis and operations of these systems. The end result is that very few people (if any) within an organization may be cognizant of all the appropriate procedures applicable to any specific system.

Diagnostic expert systems have been developed for several areas of expertise. These include medical as well as engineering diagnosis.

Recently, a number of computer-aided tools have been developed to centralize and make this procedural knowledge accessible. The tools range from simple Hypertext-based renditions of manuals to expert systems which are capable of suggesting and applying the most appropriate procedure.

In a Hypermedia document [4], information is organized nonlinearly and delivered using the most appropriate medium. This may include audio, video, text, graphics etc. The nonlinearity in the organization of the information results from the existence of multiple interrelations linking portions together. These interrelations result in easy access of related topics and avoid lengthy searches frequently encountered in a linearly organized document.

In general expert systems incorporate declarative knowledge in terms of collections of "rules" of the form

if <set of antecedents> **then** <set of consequents>

An inference engine is then used to apply rules from the rule base and advance the inferencing of the system. The order in which the rules have been entered in the knowledge base does not influence the way the inference engine proceeds, rather the rules themselves as they are applied (fired) drive the inferencing to its conclusion.

Many expert systems have been developed based on this premise, as well as several toolsets which help with the prototyping and fielding of such knowledge-based systems.

One of the most difficult problems in developing a knowledge-based system is the acquisition and coding of the knowledge of the domain experts. Knowledge acquisition involves the close collaboration of the domain expert(s) with the knowledge engineer(s). Usually, knowledge is elicited during several sessions with the participation of both the domain expert(s) and knowledge engineer(s). The thus elicited knowledge is incorporated, by the knowledge engineer in the knowledge-base, a minimal working system is rapidly prototyped and delivered. The prototype system is tested in situ by the domain expert, discrepancies and limitations are noted and the cycle repeats until satisfactory performance is obtained.

The limitation of this process lies in the difficulty of knowledge elicitation. Domain experts find it difficult to structure their knowledge in terms of general rules of the form stated earlier and to use categorical statements on exact quantities. Additionally, the knowledge engineer must at least minimally understand the domain so as to accurately code the knowledge presented by the domain expert. The existence of a widely acceptable and easily understood model for the domain knowledge, helps in its elicitation and organization.

Certain domains involve knowledge that is itself procedural in nature. Examples can be drawn from diagnosis which involves measurement or observation of specific parameters and these must follow a strict protocol either because of the expense involved, or the measurement itself is obtained as a part of a strict sequence of actions (e.g. certain blood analyses cannot be done unless the patient has fasted for a certain period of time). Physicians use protocols for both treatment and diagnosis [8]. Engineers use similarly structured procedures in operating, maintaining and diagnosing systems [2]

Most of this procedural knowledge is aimed towards human experts who are required to perform certain prescribed actions and depending on their outcome either reach a conclusion or perform another set of prescribed actions until enough information has been obtained to allow the establishment of a conclusion.

It is most often the case (especially with novices or if a procedure has not been performed recently) that the procedure, the required actions or the system itself are not familiar to the human expert. In such a case, manuals need to be consulted or the help of another expert is sought. This is time consuming, and it becomes critical in time limited procedures. Systems are currently developing which reposit such knowledge and deliver it in a user-friendly form [6].

Specifically, in the field of telecommunications and communication networks, there have been several examples of expert diagnostic systems which help interpret the status of the system and eventually produce a diagnosis.

Thus, Sugarawa [10] presented a distributed expert system capable of diagnosing local area network problems and has TCP/IP related diagnostic and troubleshooting knowledge. Miyazaki [7] describes techniques for updating switching networks operations and maintenance through expert systems. Nuckolls [9] describes an expert system that performs real time diagnosis of a large digital radio network. Yudkin [11] has introduced a methodology for building up a diagnostic system based on the structure behavior and functionality of the system and its components. In [5], we presented a preliminary system for diagnosis of a large Cable Television Network.

In this work, we shall present our efforts in developing a knowledge acquisition methodology as well as a real-time diagnostic system targeted for a large cable television distribution network. The work is divided into three parts. The first part discusses the development of FLOWTOOL^{TM}, our knowledge acquisition system, while the second part discusses the design and implementation of a real time diagnostic system capable of diagnosing a large Cable Television Network.

The current status, conclusions and future work are presented in the third part.

FLOWTOOL

FLOWTOOL comprises a graphical user interface through which the domain expert can easily represent procedural knowledge in terms of a flowchart incorporating decision points and prescribed actions. Additionally, knowledge which cannot easily be represented in the form of rules but it is also important or useful is incorporated in a hypermedia document. The user of FLOWTOOL establishes links to appropriate sections of this document. Once the acquisition phase is completed, the acquired knowledge is automatically translated to a set of rules targeted for a particular inference engine and environment complete with the established hypermedia links. The thus created prototype knowledge-based system can be utilized immediately by the domain expert who created it in the first place without the intervention of a knowledge engineer.

Operational Description

FLOWTOOL is a graphics based, flowchart editing tool which allows the user to quickly create and edit flowcharts by using a computer mouse to point and click. Flowcharts are created by dropping different flowchart symbols on a drawing area (or canvas) and connecting them together. Once a symbol has been placed on the canvas, text can be added and hypermedia links attached. At any time during the creation process, the flowchart can be edited or saved to or restored from a file. Once complete, the flowchart is checked for consistency and compiled into expert system rules.

Figure 1. illustrates the graphical user's interface involved during the operation of FLOWTOOL. The window consists of three sections, a row of control buttons along the top, four different mode buttons down the left side and a drawing area (canvas) in the middle.

The top control buttons allow the user to perform specific actions on the flowchart. They provide menus for loading, saving and editing flowcharts as well as changing the visual properties of the flowchart being displayed. The rightmost 'Flowchart' button contains menu options which let the user attach hypermedia pages and compile the flowchart into expert system rules.

The mode buttons along the side change the current state of the program. The current state determines what action will be taken when the user presses the mouse button on the canvas. While on the canvas, the cursor will change shape to reflect the chosen mode.

Flowchart symbols are drawn by first clicking on a symbol mode button (i.e. a statement or decision button) and then clicking on the canvas. When the mouse pointer is moved back over the canvas, its shape will change to either a rectangle (statement) or a diamond (decision), depending on the symbol to be placed. Whenever a new symbol is dropped, it is highlighted, designating it as the current symbol and a dialog box will appear to display information about it. The sym-

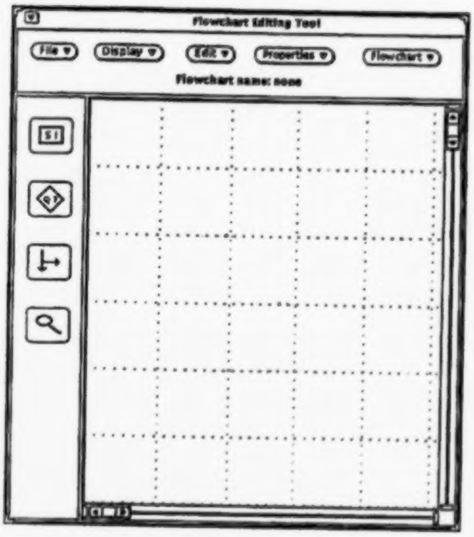


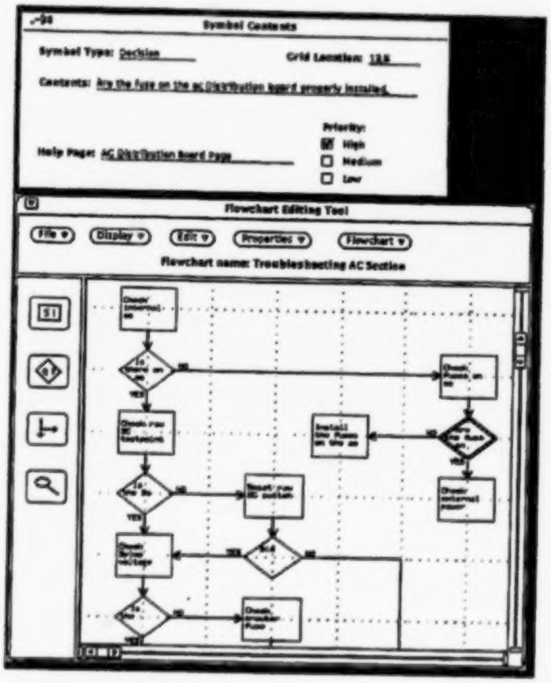
Figure 1: FLOWTOOL's graphical user's interface

bol's dialog box displays the symbol type, its location on the grid and its textual contents. In addition, if a hypermedia page has been attached, then the name of that page and its priority are also displayed.

Once two or more symbols have been placed, they can be connected together to form a directed arc in the flowchart. To connect two symbols, the user must first click on the connect mode button. The user is then prompted to identify source and destination symbols and, having done this, a link is then created between them.

The contents of any symbol can be displayed by clicking on the examine mode button. This causes the cursor to change shape to a magnifying glass, denoting the current mode. When this magnifying glass is then moved over any symbol, a dialog box will appear, displaying the contents of that symbol.

Figure 2. shows a completed flowchart including the contents of a flowchart symbol. The 'Symbol Contents' dialog box displays information about the highlighted decision symbol at the right of the canvas. Besides the full contents of the symbol, a hypermedia page entitled 'AC Distribution Board Page' is also shown with a 'high' priority. This means that when this node in the flowchart is encountered during flowchart traversal, the designated help page will appear (if the user has chosen to accept high priority help pages).


Symbol Contents

Symbol Type: Decision

Grid Location: 13,6

Contents: Are the fuse on the ac Distribution board properly installed.

Help Page: AC Distribution Board Page

Priority:
☑ High
☐ Medium
☐ Low

Figure 2: Example of Procedural Knowledge Acquisition.

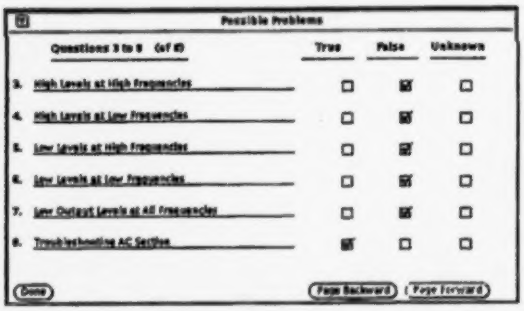
At any point, the flowchart depicted, can be compiled to a cluster of rules encapsulating the procedural knowledge depicted. The compilation process is straight-forward. Each decision box results into two rules, one for the affirmative and the other for the negative outcome. The contents of a decision box together with references to its preceding decisions constitute the antecedents of the rule, while the statement(s) following a decision path constitute the consequents of the rule.

Additionally, any links to hypermedia pages present, are attached to the rule(s) created. A hypermedia server intercepts requests to display the attached hypermedia pages when the rule fires.

A flowchart incorporates procedural knowledge pertaining to diagnosis, calibration, operation etc.

Once a flowchart has been compiled, the resulting cluster of rules joins the collection of other clusters to form a knowledge base of procedures normally associated with a particular system. At any instant, one or more of these procedures may be applicable. For example, in diagnosis, each cluster would normally be associated with a specific manifestation of a malfunction. When the user wishes to use the expert system to solve a problem, he first describes the problem in terms of the observed "behavior", in the case of diagnosis by enumerating the observed symptoms from a list of known symptoms.

This is done using the 'Possible Problems' dialog displayed in Figure 3.

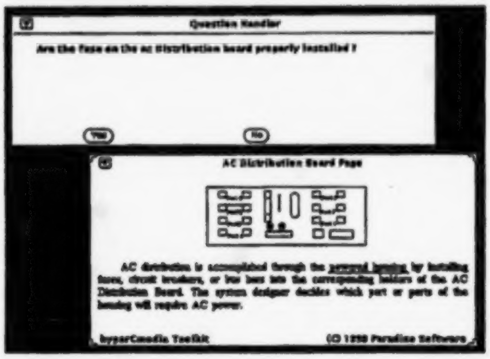


Questions 3 to 8 (of 8)	True	False	Unknown
3. High Levels at High Frequencies	☐	☒	☐
4. High Levels at Low Frequencies	☐	☒	☐
5. Low Levels at High Frequencies	☐	☒	☐
6. Low Levels at Low Frequencies	☐	☒	☐
7. Low Output Levels at All Frequencies	☐	☒	☐
8. Troubleshooting AC Section	☒	☐	☐

Figure 3: A possible-Problems Screen

Any listed problem checked true will cause the associated cluster to be processed by the expert system. Problems that are checked unknown will also be processed, but only after those checked true have been examined.

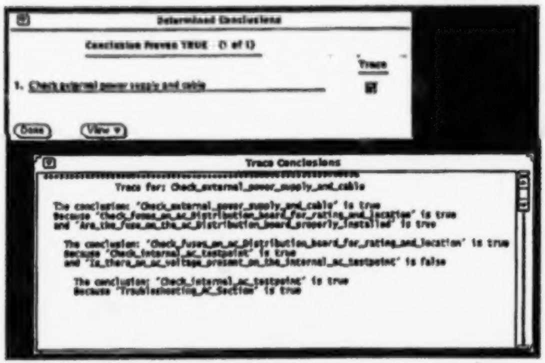
When the user hits 'Done', all of the requested flowcharts will be processed. Any hypermedia pages attached the flowchart symbols will also appear. Figure 4 shows how the user is prompted to answer a question from the 'Troubleshooting AC Section' flowchart. A hypertext page appears with the question to help the user make the correct response.



Are the fuse on the ac distribution board properly installed ?

Figure 4: An inquiry presented during the operation of the expert system obtained from procedural knowledge capture by FLOWTOOL. Observe the associated hypermedia page which is also presented.

Once the user has answered all questions leading to a conclusion, another dialog appears to display the results of having navigated the knowledge base. If desired, the user can also view a trace of flowchart just processed. This trace details the decisions that were made in processing the flowchart and shows how the conclusions were arrived at. This is shown in Figure 5.



Conclusion Proven TRUE (1 of 1)

	Trace
1. Check external power supply and cable	☒

```
Trace for: Check_external_power_supply_and_cable

The conclusion: 'Check_external_power_supply_and_cable' is true
Because 'Check_fuses_on_ac_Distribution_board_for_rating_and_location' is true
and 'Are_the_fuse_on_the_ac_Distribution_board_properly_installed' is true

  The conclusion: 'Check_fuses_on_ac_Distribution_board_for_rating_and_location' is true
  Because 'Check_internal_ac_testpoint' is true
  and 'Is_there_an_ac_voltage_present_on_the_internal_ac_testpoint' is false

    The conclusion: 'Check_internal_ac_testpoint' is true
    Because 'Troubleshooting_AC_Section' is true
```

Figure 5: Typical Conclusion and Trace Screens

Network Diagnosis

Structure of the Network

A Cable Television Network incorporates a number of high frequency amplifiers forming (for conventional networks) a tree. In more advanced networks the structure incorporates a double ring from which subscriber drops emanate. In this work, we are focusing in conventionally structured tree networks. There are two categories of amplifiers, the ones belonging to the main trunk and the ones forming subscriber drops. Additionally, power supplies are located throughout the network, each one powering a limited number (typically three) of amplifiers. The majority of the main trunk amplifiers are equipped with a status monitor which uses a reverse channel to report the status of the amplifier to the head office. Subscriber drops and power supplies are not normally monitored.

Typical variables which are monitored include [1]

- Output pilot level
- Output data carrier level
- Raw DC voltage into the amplifier
- B+ voltage of amplifier power supply
- DC current into forward and reverse sections of the amplifier
- Temperature inside the trunk station
- Reverse Switch status
- Trunk lid status

The values of the monitored variables are allowed to vary within two intervals (warning and alarm) centered at typical values. If a value is outside these predefined intervals then a warning or an alarm is issued. Three consecutive alarms constitute a failure. Failure to raise a particular status monitor for three consecutive time intervals, also constitutes a failure.

Each status monitor apparatus has its own electronic address, which is used at the head end to poll it for a sta-

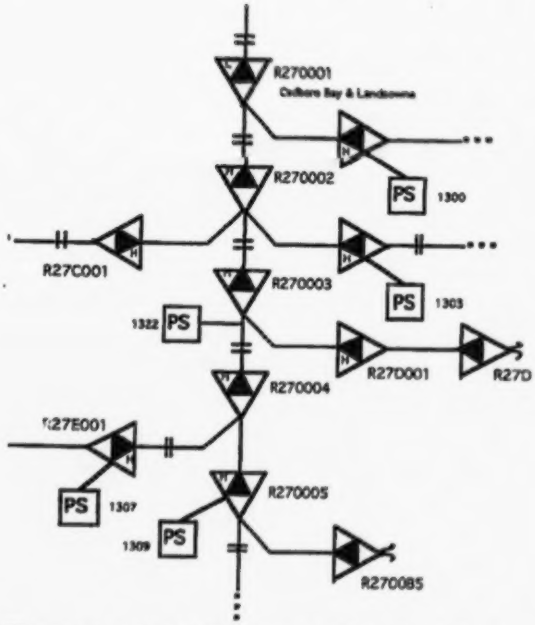


Figure 6: A typical Section of a Main Trunk (curtesy Roger's Victoria)

tus report.

Because of the number of amplifiers in the network, each amplifier is polled at fixed intervals (typically every few minutes).

The polling order may change if a particular section of the network needs closer attention.

A typical section of the main trunk is depicted in Figure 6. Each amplifier in the network has a name, as well as a location, connectivity and functionality attributes.

Diagnosis Problem

There are several modalities of failure. Some are discussed below.

A single amplifier may fail, whereupon the signal is fed through unamplified to the subsequent stage of the network. The subsequent amplifiers, equipped with automatic gain control, will boost the signal back to its normal level after two to three stages. Because of the failure, all amplifiers located between the failing amplifier and the stage at which the signal was boosted to its correct level, report alarms or failures. Because of the tree structure of the network, the reporting amplifiers are not typically polled sequentially and the reports appear at seemingly random locations in the report.

A power grid failure, will affect a number of amplifiers. The affected amplifiers fail to communicate, while the deteriorated signal, cause downstream nodes to report failure until the signal is boosted again to its typical values. Again, because of the structure of the network, these reports appear at random locations and are seemingly unrelated.

Another category of failures are due to the unmonitored subscriber drops. These are typically reported by the subscribers themselves as deterioration of service which range from increased noise levels for some channels to complete interruption of reception. These reports are due to failures in the main trunk, subscriber drops or even interruption of service due to the customer being at arrears in his/her payments.

In this work, we shall concentrate on the diagnosis problem of the monitored main trunk.

Diagnosis Approach

We follow a model-based approach in our reasoning. Because the structure of the network is known a-priori, and since the behavior of the amplifiers in the network is well understood, and because there is a large number of elements in the network, model-based reasoning is best suited for our domain. Model-based reasoning, as opposed to classification reasoning, bases its decisions on an underlying model of the system being diagnosed.

In our case, the structure, and operation of the elements in the network, provide us with a general and powerful model on which we base our diagnosis.

As we discussed previously, the failure is localized to at most a small number of amplifiers, but because of their interconnection, deteriorated levels of the signal propagate and affect a large number of neighboring nodes which themselves report failure. One of the main objectives of our diagnostic system is the identification of the *truly-failed* nodes from the ones which report failure because of their proximity to the failed nodes. Such agglomerations constitute *failure clusters.*

A *failure cluster* is defined to be a connected set of nodes, at least one of which exhibits a failure while the remaining register a warning, alarm or failure within an observation window which spans several observation cycles. For the remaining of the discussion, we shall use the term *failure cluster* to denote both a set of amplifiers as defined above, and its representation.

Failure clusters are important because they incorporate all the affected amplifiers within an observation window, and the reasoning engine can focus only within the cluster. Additionally, forming the failure clusters dynamically, we isolate the reasoning process from sources of information which cannot be made temporally constant. For example, the amplifier data-base together with the connectivity information changes in time as the cable-plant evolves.

Failure clusters act as filters, preventing the continuous formation of diagnoses of the same problem that has been previously diagnosed and is currently awaiting resolution.

Based on the above observations, we have constructed a system which automatically and in real time receives the status monitor data from the amplifier network and produces a diagnoses in cases of failure. As depicted in Figure 7. above, our system comprises a data server, a compression and archival module, the cluster generator and the diagnosis module.

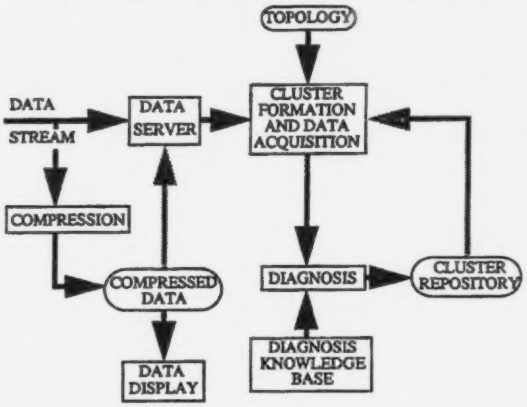
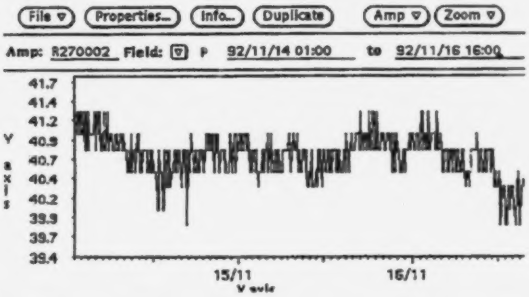


Figure 7: The structure of the Expert Network Analyzer

The stream of the status monitor data is captured, compressed and archived. The archiving is important, since this data is to be used in further analysis and failure prediction. A specialized display tool has been written which is capable of displaying arbitrary sequences of status monitor data. A typical display of such a sequence is depicted in Figure 8. below.



Amp: R270002 Field: ▽ P 92/11/14 01:00 to 92/11/16 16:00

Figure 8: Display of a typical status monitor data sequence. This pertains to the pilot level.

A Data Server obtains data either directly from the status monitor stream or from the compressed and archived data and sends them to the cluster formation and data acquisition module. There, new clusters are formed according to the algorithm presented in Figure 9. The seed amplifier is defined as an amplifier that reports failure and does not belong to any of the currently active clusters. Once a cluster has been formed, it is sent to the diagnosis module which analyzes it and produces a diagnosis of the failure. This diagnosis is currently *mailed* electronically to the domain expert who provides a *critique* of the diagnosis conclusions reached. These critiques are saved and will be used to validate the diagnosis knowledge base.

Finally, the diagnosed cluster is deposited to the "active cluster" repository and remains there until the fault is repaired and all the amplifiers report normal operation.

Failure clusters, in addition to the amplifier names, incorporate connectivity information as well as observation information within the window of observation.

```
get next status monitor data
name the amplifier It
determine Its status

case (Its status is OK):
  if It is in queue[i] discard It from queue[i]

case (Its status is failing ORin alarm OR in warning):
  if It is in queue[i]
  then
          put It in newcluster[i]
          discard It from queue[i]
          put Its children in queue[i]

case (Its status is failing):
  if It is not in any of the queues
          if It is not in any of the clusters
              It is the seed of a new cluster
              make a new cluster
              make a new queue
              put It in the just formed newcluster
              put Its children in the just formed queue

if queue[i] is empty
  export newcluster[i] for data collection & diagnosis
  delete newcluster[i]
  delete queue[i] .
```

Figure 9: Cluster Generation Algorithm

A failure cluster is translated into a set of NEXPERT objects which fully quantify the topology and behavior of the cluster within the window of observation. Each amplifier in the cluster comprises three objects, namely the topology_data object, the amp_data object and the log_data object. The topology data object incorporates static properties of the amplifier such as the location, connectivity, power grid, status monitor number, termination etc. The log data object incorporates the fault data of the amplifier such as the fault message, the fault type (warning, alarm, failure) the actual and nominal values of the variable that has caused the failure and the fault times. Finally the amplifier data object incorporates the observed data for all the monitored variables of the said amplifier within the window of observation.

Typical examples of objects are given in Figure 10., Figure 11. and Figure 12.

The set of objects, corresponding to a failure cluster and obtained as discussed above, is exported to the rule based system which produces a diagnosis. The rule based system, implemented in NEXPERT OBJECT, currently comprises over 80 rules and is capable of diagnosing 12 different failure modalities. A preliminary evaluation of the diagnoses, has established an accuracy rate of approximately 80%.

```
(data_R20000D
  Classes: amp_data
  Properties:
     B     (23.8,23.8,23.8)          /* B+ voltage */
     C     (1143.3,1143.3,1143.3)   /* current */
     P     (-100.0,-100.0,-100.0)   /* forward pilot */
     R     (-6.0,-6.0,-6.0)         /* reverse pilot*/
     RD    (54.9,54.9,54.9)         /* Raw DC voltage */
     T     (-10.0,-9.0,-10.0)       /* Temperature */
     times (12140916,12140919,12140922) /* Pol times */
)
```

Figure 10: Example of an amplifier data object

```
(topol_R20000D
  Classes:      topology_data
  Properties:   location        (Haultain@foulbay)
                parent          (R20000C)
                pilot           ()
                power_grid      (1209)
                SMT_number      ()
                subs            ()
                termination     (False)
                type            ()
)
```

Figure 11: Example of a topology_data object

```
(log_R20000D
  Classes:      log_data
  Properties:   fault_message  (Communication)
                fault_nominal  ()
                fault_times    (12140931)
                fault_type     (Alarm)
                fault_value    (NO REPLY)
)
```

Figure 12: Example of a log_data object describing an amplifier in an alarm condition because of failure in replying to a polling request.

Status and Conclusions

The work described here is part of a larger undertaking in collaboration with the Canadian Cable Labs Fund to investigate the use of AI and Knowledge Engineering techniques in the automation of the diagnosis procedures in a large cable TV plant. The development has proceeded in two phases.

During the first phase, we have developed FLOWTOOL™ a graphical knowledge acquisition tool which is capable of capturing knowledge expressed in the form of flowcharts and translating it into rules for NEXPERT OBJECT™. In addition, the resulting knowledge base is linked to hypermedia documentation incorporating information that cannot be expressed in the form of rules within the knowledge base. Appropriate sections of this documentation is being made available automatically during the operation of the inferencing and this can be used to assist and/or train the user of the knowledge base in providing appropriate responses and comprehending the inferencing involved.

The second phase includes the development of the Expert Network Analyzer.We have completed the beta version of our expert network analyzer on a SPARC platform in C and NEXPERT OBJECT. As discussed, this version of the expert network analyzer operates on line and in real-time. We are in the process of validating and refining the analyzer. This is done through the critiques of the diagnoses which are *electronically mailed* to the domain experts.

One more area that we plan to enhance our system, is its responsiveness. Currently, all the amplifiers in the network are polled sequentially, each complete polling sequence requiring approximately 90 sec. In order to form a failure cluster and accumulate enough data for a diagnosis to be possible, several polling cycles (currently 5) are required. This results to a delay of approximately 8 min from the onset of a failure to the time that a failure cluster is formed and a diagnosis becomes possible. In order to enhance the response time, we are currently implementing failure directed polling. This means that in addition to the standard sequential polling of all the amplifiers in the network, our system will be capable of polling specific amplifiers involved in the formation of failure clusters out of sequence and very rapidly. This would have as an effect the minimization of the delay in forming a failure cluster, and thus the generation of the diagnosis soon after the onset of a problem.

Lastly, we are currently focussing our attention in developing failure predicting models. For this part of the work, we utilize the archived status monitor data of the entire Roger's Victoria network, which we shall use in developing our predictive models. The status monitor data have revealed some interesting patterns of behavior, which we hope to correlate with and use as precursors of failure. Such patterns include the variation of the forward pilot level in close correlation with the temperature of the amplifier itself, as well as seemingly random variations of the current. Such patterns are observed only for data corresponding to failing amplifiers, while amplifiers operating normally exhibit stable readings for very long periods of time.

Acknowledgment

The support of the Canadian Cable Labs Fund is gratefully acknowledged.

References

1. *Instruction Manual and User Guide Status Monitor System*, C-COR, 5/84, rev 0, May, 1984.
2. Preliminary Instruction Manual for the B-507 Remote Bridger C-COR Inc. July 1983

3. S. Abu-Hakima "Visualization and Understanding Diagnoses" *Canadian Artificial Intelligence* No. 30 pp. 4-8 (Autumn 1992)

4. J. Conklin "Hypertext: An Introduction and Survey" *IEEE Computer* Vol 20. No. 9 pp.17-41 (Sep. 1987)

5. N. J. Dimopoulos, K. F. Li, A. Watkins, S. Neville, A. Rontogiannis "An Expert Network Analyzer" *Proceedings of the 35th Canadian Cable Television Association Annual Convention* pp.123-127 (Jun. 1992)

6. P. R. Frey, W. B. Rouse, R. D. Garris "Big Graphics and Little Screens: Designing Graphical Displays for Maintenance Tasks" *IEEE Trans. Systems Man Cybern.* VOL. 22, No. 1, pp. 10-20, Jan/Feb. 1992

7. Miyazaki, T., Fujimoto, H., Kim, M.W., and Wakamo, M., "Improving Operation and Maintenance for switching network," in *Proceedings of GLOBECOM '89*, IEEE, Nov. 1989, pp. 1149-1153.

8. M. A. Musem, L. M. Fagan, E. H. Shortliffe "Graphical Specification for an Expert System" in J. A. Hendler (Ed.) *Expert Systems: The User Interface* Ablex Publishing Corporation, Norwood, New Jersey.

9. Nuckolls, V., "Telecommunications diagnostic expert system," in *Proceedings of GLOBECOM '89*, IEEE, Nov. 1989, pp. 507-511.

10. Sugawara, T., "A Cooperative LAN Diagnostic and Observation Expert System," in *Proceedings Ninth Annual International Phoenix Conference on Computers and Communications*, IEEE, March 1990, pp. 667-674.

11. Yudkin, R.O., "On Testing Communications Networks," *IEEE Journal on Selected Areas in Communications*, vol. 6, no. 5, 805-812, 1988.

12. A. Watkins, N. J. Dimopoulos, S. Neville, K. F. Li "Flowtool: A Procedural–Knowledge Acquisition Tool" *Proceedings IEEE Pacific Rim Conference on Communications Computers and Signal Processing* pp. 31-34 (May 1993)

Y25021

Distribution Planning: A Constrained Heuristic Search Approach

Victor Saks, Ivan Johnson
Carnegie Group, Inc., 5 PPG Place , Pittsburgh, PA 15222
tel: +1-412-642-6900; fax: +1-412-642-6906; internet: saks@cgi.com

#149609

Mark S. Fox
Department of Industrial Engineering, University of Toronto,
4 Taddle Creek Road, Toronto, Ontario M5S 1A4
tel: +1-416-978-6823; fax: +1-416-978-3453; internet: msf@ie.utoronto.ca

Abstract

KBLPS is a state-of-the-art Decision Support System with a rich interactive graphics user interface, advanced knowledge base and powerful planning algorithms that has been designed and implemented to support skilled U.S. Army logisticians to prepare and evaluate logistics plans to support corps-level battle scenarios much more rapidly than currently possible. At the heart of KBLPS is the commodity Distribution Planner (DP) algorithm, which in a single framework supports Ammunition Distribution and Petroleum Distribution planning. KBLPS uses the problem solving model of Constrained Heuristic Search (CHS). CHS views the construction of the distribution plan as a constraint optimization problem and uses graph textures to guide and control the problem solving. KBLPS is fully implemented in C++ and has undergone a significant unit and integration testing, and is currently being used by the USArmy.

1.0 Introduction

The complexity of military logistics planning and emerging needs for ever-higher degrees of responsiveness motivate the requirement for automated decision support tools that complement the strengths of the human logistics planner. The Knowledge Based Logistics Planning Shell (KBLPS) under development by Carnegie Group, Inc. (CGI) and LB&M Associates with the sponsorship of the Combat Service Support Division of the U.S. Army's Human Engineering Laboratory (HEL) and the U.S. Army Strategic Logistics Agency (SLA) provides such support tools.

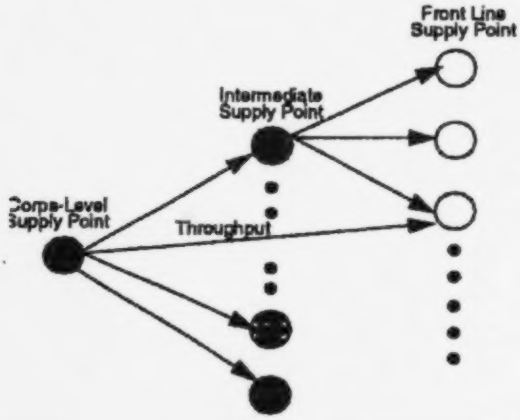
At the heart of KBLPS is the Distribution Planner (DP) planning algorithm, which underpins the Ammunition Distribution Planner (ADP) and the Petroleum-Oil-Lubricant (POL) Distribution Planner (PDP). A major challenge has been to design a single common algorithm general enough for use by these two domain-specific planners. The ADP and PDP support CORPS/G4 Cell planners in assessing the supportability of proposed Maneuver Courses of Action (MCoA) from ammunition and POL logistics perspectives. The DP algorithm reasons and calculates at appropriate levels of aggregation to demonstrate the logistics supportability of a MCoA without overwhelming the user with too much detail.

KBLPS and the Logistician can work together to generate commodity distribution plans much more rapidly and with greater accuracy, efficiency and responsiveness than now possible for the Logistician working alone. The Logistician can configure the problem/scenario and give guidance to the algorithm by optionally setting various parameters; the algorithm constructs a distribution plan which involves significant computational complexity. As a consequence, the Logistician can spend more time analyzing and assessing plans than generating them.

In the following, we describe the logistics planning domain in more detail, followed by a description of the problem solving method based on Constrained Heuristic Search [Fox et al., 1989]. Lastly, we describe the development process, observations and current status.

2.0 Problem Description

The DP addresses the problem of supporting a corps-level combat mission maneuver course of action by planning to move as much product as possible forward to the forward supply points, to meet the user-unit product consumption/demand over a user-specified time period (typically 3-5 days in this domain).



The above picture depicts the flow of materiel. Ammunition distribution requires the planning of ammunition shipments from Corps Support Areas (CSA's) to Ammunition Supply and Transfer Points (ASP's & ATP's); POL distribution plans POL shipments from General and Direct Support (GS, DS) points to Main and Forward Support Battalions (MSB's & FSB's). Each domain has particular aspects which make it complex. For example, there are far more Ammunition types (called DODICs) than POL product types. On the other hand, POL distribution resources such as tankers and storage bags are product specific, which is not the case for ammunition. The shape of the distribution network is potentially more complicated for POL.

The principal goal of distribution planning is to maximize the satisfaction of unit demand. Priorities must be respected so that higher priority orders are satisfied before lower priority orders. But the preference for higher priority orders must be tempered by the objective that lower priority orders cannot be completely ignored and starved; hence, there must be a balance between these conflicting goals.

An important secondary objective is to meet day-to-day inventory stockage objectives, which are Logistician-specified in terms of number of days of supply in the context of the CoA being supported. When in conflict with the primary objective, this objective is relaxed.

The distribution planner must assign Material Handling Equipment (MHE), Main Supply Routes (MSR), truck and available inventories to supply the user units consumption/demands in the best possible way over time. In general, this is an "over-constrained" problem - the net demand is generally substantially greater than inventory and/or material handling and carrying capacity can handle, at least in some time-frames. Hence, some of the demand will remain unsatisfied. It is the G4 Log Planner's responsibility, to find the means to move more material and/or to set aside the subset of demand that can best be "left aside" and still meet the commander's intent with the subset of materials that can be moved.

Moreover, in different scenarios different resources will be most scarce. Thus, the planner must maximize efficient usage of all resources, and in particular the scarce resources, in order to maximize over-all order satisfaction. This must be done in light of commander intent/mission, balanced servicing of all combat units so they can adequately support the mission and each other, and available assets.

3.0 Overview of Approach

The planning algorithm plans commodities distribution through its model of the distribution network of supply points, with the objective to maximize on-time satisfaction of demand, in the form of orders, and meet mission-driven stockage objectives. The generated plans are feasible - resource capacity constraints are not exceeded at the level of aggregation being used.

The DP algorithm constructs a plan by iteratively selecting an order (the 'next best' order not yet in the plan) and then placing that order into the plan. The DP reasons and plans "opportunistically", in the sense that neither the sequence in which orders are planned nor the way in which orders will be planned are determined in advance. Since there are typically a large number of orders which could be planned next and numerous ways to plan them, the DP uses *knowledge of the problem structure* to guide the search to make these decisions quickly, efficiently, and accurately. More precisely, high priority orders which rely on resources which are heavily contended for in certain time intervals are planned as early as possible in the planning process, to ensure that these high priority orders are not left out of the plan. The algorithm strives to relieve contention for over-constrained resources by planning elsewhere (in time and/or location) orders that are contending for those resources and that also have reasonable alternatives.

The DP algorithm provides a mechanism which allows a Logistician-specified percentage of higher priority orders to be satisfied first, then a percentage of lower priority orders, and then the remainder of the higher priority orders. In this way, priority is respected but lower priority units and orders are not starved completely. This balance mechanism gives control to the Logistician, and is consistent with the KBLPS philosophy of being a Decision Support System.

3.1 Constrained Heuristics Search

KBLPS uses the problem-solving model of Constrained Heuristic Search (CHS) [Fox et al., 1989]. CHS combines the power of Constraint Satisfaction (CSP) techniques with heuristic search. It uses a constraint graph to model a problem and uses measurements of the properties of the constraint graph, which we call textures, to guide the search process of iteratively selecting a variable and assigning a value to it.

CHS models a logistical problem using a temporal/capacity constraint graph (figure 1). The nodes are variables to which one or more values are to be assigned. In the logistics domain, nodes in the graph represent activities to be scheduled, i.e., assigned a start time, end time, and a set of resources. The arrows represent temporal constraints between activities that the assignment of the start and end times must satisfy (e.g., precedence constraints). In this case we show only "before" temporal constraints. The heavy arcs are capacity constraints. They are disequalities that specify that two connected activities cannot use the same resource at the same time, i.e., cannot be assigned the same resources during the same temporal period.

An important component of the CHS model is that there exist domain independent properties of the constraint graph, called textures, that can be used to construct search heuristics:

Value Contention: degree of to which more than one variable wish to have the same value.

Variable Reliance: degree to which a variable relies upon the assignment of a particular value.

Variable Looseness: size of range (conjunction of constraints).

Constraint Tightness: degree to which the constraint reduces the set of admissable solutions.

Constraint Importance: how important is it to satisfy the constraint.

Figure 2 depicts a *Contention Graph*, which is a elaboration of the temporal/capacity constraint graph. It adds an additional node for each resource so that texture measures can be stored in the graph at the resource nodes. The picture depicts the aggregate demand over time that exists for Resource R_3 as imparted by the individual demands of Activities A_{13}, A_{21}, and A_{23}

Problem solving is performed by iteratively selecting an activity and assigning a temporal period, i.e., start time and end time, and one or more resources. In particular, the following steps are followed cyclically until all activities have been assigned resources and time

Figure 1.

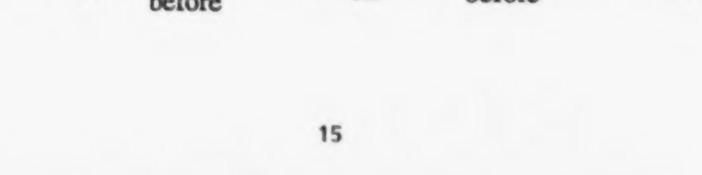
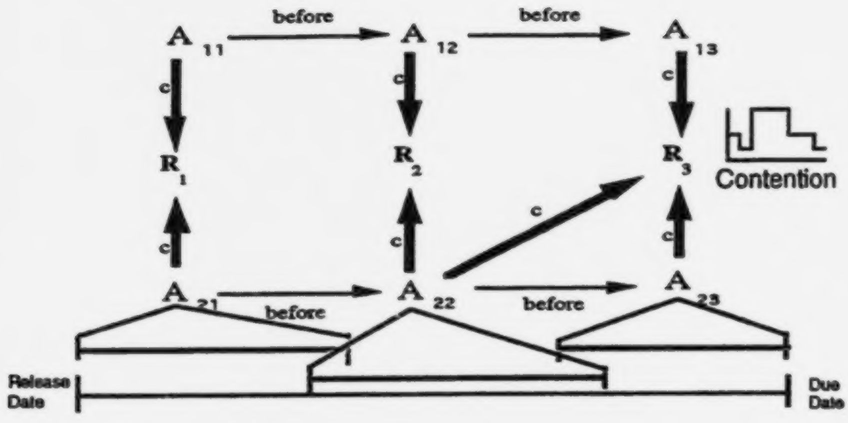


Figure 2.



periods. Given a logistics problem modelled using a contention graph:

1. Perform constraint propagation (aka arc-consistency).
2. Measure textures.
3. Select an activity.
4. Assign a temporal period and resources.

In the context of distribution planning, textures that are measured are contention for a resource from activities, and the reliance an activity has on a particular resource. The DP algorithm first identifies which resources are most heavily in demand relative to their availability at specific time intervals within the planning horizon, and then assigns them to the most reliant activities so as to exploit them as completely as possible.

Backtracking occurs whenever a deadend is reached. Experience with this approach in the domains of factory scheduling and spatial planning have shown that backtracking is minimal [Sadeh, 91; Baykan & Fox, 92].

3.2 Algorithmic Steps

KBLPS implements a variation of CHS based on the work of [Sadeh & Fox, 90; Sadeh 91]. For each order for ammunication or POL, there exists more than one way of satisfying it. A chain of activities defines the 'process steps' that must be performed to move material forward; these include truck loading, hauling a quantity of commodity from one supply point to another, etc. Each order can be satisfied by more than one chain of activities, resulting in a tree of alternative activity sequences. Activity Model 'templates' define these generic process/activity sequences. The union of these activity tree templates for each order defines the constraint graph for the distribution planner.

At this point, the constraint graph is not yet complete. The demand an individual activity has for a resource is not yet known in the activity tree template for an order. The demand for ammunition (or bulk petroleum) is probablistically propagated through the distribution network, thereby distributing an individual order's demand across the alternative activity chains that may satisfy it.

With the constraint graph now complete, textures can be measured. The impact on the modeled resources (trucks, roads, MHE) if the demand were to be totally satisfied is calculated. The need for the resources is compared to their availability in 'time/resource chunks'. The resource(s) most in demand relative to their availability are singled out for scrutiny. The demand (in the form of back-propagated orders) contending for those resources are reviewed for criticality; those orders most dependent (they have fewest alternative ways of being satisfied) on those resources are selected to be placed into the plan. This is done in such a way, however, (e.g., by 'sliding' them in time) as to relieve the bottleneck effect on the heavily-contended resource(s) as much as possible. The remaining (as yet un-scheduled) part of the problem is then re-assessed and the next-most-bottlenecked resources and most-critical orders are dealt with. This cyclical process continues until there is no remaining demand to satisfy or there are no time/resource fragments remaining that can be used to build another plan segment.

Of course, the process is considerably more complex in detail than this summary might suggest. For example, at the first step, propagation of demand through the distribution network, a probabilistic view is imposed, recognizing that there is uncertainty in exactly when material movements will actually be assigned, how long shipment will take, which orders and associated quantities will succeed in capturing resources, etc.

The basic sequence as the Plan is generated is: (1) Measure Textures; (2) Select an Order; (3) Plan the Order; (4) Record the plan fragment. We discuss these steps here in more detail:

1. *Measure Textures:* For each activity, determine its probablistic demand for each resource it needs, such as ammunition, material handling equipment, etc. For each resource, aggregate the probablistic demand from each activity that needs it. For each activity, determine its reliance on each resource it needs.
2. *Select Order*: Select the (approximately) most-contended-for resource in some specific time interval. At this point a few or many resources may be approximately equal in their contention measure. Being able to control how approximate this equality measure is gives us the capability to trade off some detailed precision for execution speed; the algorithm can select one of the approximately most-contended for resources much more rapidly than selecting the exactly most-contended for resource. (An underlying observation here is that the contention measures are probabilistic, and hence not exactly precise.) We are continuing to explore how sensitive plan results and computation speed are to these approximation factors. Numerous researchers in planning and scheduling have discussed order selection. We have been influenced most by [Fox et al., 1989; Sadeh & Fox, 1990; Sadeh, 1991]

   The DP algorithm then selects an order from the group of propagated orders contending for the selected highly-contended resource at that time. This selected order has approximately the highest "Reliance" on the critical resource in that time interval. High Reliance reflects the order's need to be planned on the critical resource in that time interval; from a planning perspective, it is important to make that decision now (i.e., on this plan fragment cycle) before the critical resource is consumed by another order which doesn't need it as badly. The algorithm also works to plan the order without using the critical resource in the critical time interval, thereby helping to relieve that bottleneck. Moreover, the specific activity (e.g., unload trucks) performed on the critical resource (e.g., limited MHE) is the focus for planning the order; we refer to this as the "critical activity" (CA).
3. *Plan the Selected Order*: Planning the CA is the focus of planning the order. First, the algorithm constructs a feasible time window for the CA. This is done by propagating backwards through the order's Activity Model from the order due time and forwards from the earliest feasible time for beginning the shipment (time is represented continuously). The CA can be laid into any (sufficiently long) time block within this window; the order will be feasible with respect to all of its other activities.

   The algorithm works to choose the start time for the critical activity which reduces the contention for the critical resource to the maximum extent possible, using the contention measures discussed above. The algorithm then proceeds to plan up- and downstream activities for the order with a just-in-time approach. Each order is planned such that its execution is feasible; hence, the plan is feasible at each successive step (cycle) of the plan-building process; at whatever point the data drives the algorithm to stop processing, the plan will be feasible. There may be some demand that remains unsatisfied as resources have been consumed by higher priority demand, but the overall plan as generated will be feasible according to resource availabilities.
4. *Record Plan*: Having laid another plan fragment into the overall emerging plan, the algorithm updates the resource consumption/availability tables and the demand profiles for resources which have been affected. Some constraint propagation is also performed at this point as needed; e.g., when resources become exhausted in some time interval, demand for those resources is recalculated for the time intervals where they have not been totally consumed.

4.0 Development Process

The developmental process has gone through several stages of approach and intensity during early requirements definition and more recent focused development:

- Conceptual definition of logistics planner tasks, and what subset could be reduced to engineered software solution(s); this was accompanied by a series of early (LISP-based) prototypes to help clarify requirements, needs, and possibilities; these efforts led to first versions of the DP algorithm outside of a basic research environment.
- First fully integrated KBLPS version, using scanned paper maps, limited (notional Xth corps) battle scenarios; expanded second version (applied research) of DP algorithm; still largely LISP-based environment.
- Current fully integrated (fully C/C++) version; software engineering methodology (using CASE tools) to develop firm design requirements, coding and testing of GUI, KB, and DP and Transportation Scheduler algorithms; step-wise unit testing and module integration; exhaustive stand-alone algorithm testing, refinements, and changes to increase execution speed; exhaustive integrated testing; current on-going field testing with skilled world-class logisticians.

  The current version of KBLPS has been tested on problems of a maximize size of 4,000 orders, 20,000 activities, and 300 resources. The average execution time for a problem with 2,000 orders and 10,000 activities is 2.5 seconds on a Sparc10.

5.0 Validation Process

In order to facilitate testing and validation, we developed a comprehensive Test-Case Generator (TCG), which takes a high level scenario description as input and outputs a full scenario description which can be input into the DP. We first generated small simple scenarios, gradually upgrading the TCG so that it now can build scenarios that stress the algorithm in specific and controllable ways. The TCG was indispensable as we debugged and improved the DP; we now have a test suite of over 100 test cases.

It is useful to distinguish between bugs that cause the system to crash and bugs which do not. The former are usually much easier to find, isolate the cause, and repair. The DP algorithm is complex; it was inevitable that there will have been design and coding errors which do not cause the system to crash, but do cause the algorithm to make poor decisions. These bugs are much harder to find, isolate the cause, and fix. After a substantial baseline algorithm evaluation and debugging process was completed, the DP was reliable in terms of running to completion even on completely new scenarios. We then spent a second significant amount of time examining DP results for plan quality, i.e., looking for situations where the DP made a poor decision, figuring out why, and improving the algorithm to make a better decision in that situation. Unfortunately there really is no alternative to this labor intensive method.

The algorithm subsequently went through a thorough examination by the domain experts at LB&M Associates. As a result, they were quite impressed with the algorithm's responsiveness and sensitivity to changes in the model, so that, for example, if a little more resource is added, the algorithm predictably plans more orders, and vice versa.

The result of this painstaking debugging process is that the DP is quite robust, running successfully to completion on new scenarios and generating high quality plans.

6.0 Observations

The design of the DP algorithm has been challenged by the need to consider carefully the trade-off between plan quality and execution speed. Too much calculation can take too long, while too little can result in lower quality plans. This issue has been noted by other researchers trying to build a real system with users in the field [Sadeh, 91; Zweben et al., 90]. As we gain more experience with the algorithm's behavior and the utility of the plans it produces, we anticipate design refinements to continue to balance speed vs. quality. One aspect may well be other layers of aggregating data (e.g., placing groups of ammunition types into a single "demand bucket" thereby reducing the data-intensity of a problem scenario.)

We have succeeded in designing a single algorithm capable of handling both Petroleum and Ammunition commodity distribution. There are some key differences between these, perhaps most notably the heavy reliance on Throughput (direct ammunition trans-shipment from CSA to ATP) and less reliance on Throughput with Class III. KBLPS enables the logistician to specify the mix of throughput and supply point-to-supply point shipment he prefers to aim for in any particular scenario. The DP algorithm respects those user-specified preferences and works toward satisfying orders accordingly.

The DP algorithm is designed to be opportunistic, so that while it is constructing the plan, its selection of the most constrained resource on which it focuses its attention is completely dynamic. For each of the modeled resources, we have test case scenarios in which that resource is clearly the most constrained, and the DP continuously focuses its attention on that resource. In other scenarios, the are several resources which have heavy contention and the DP shifts its focus from one to the other, so that as it finishes planning one bottlenecked resource, it shifts its attention to another.

The DP is also designed to relieve bottlenecks, so that sometimes as the DP actually relieves a bottlenecked resource by planning orders elsewhere, that resource becomes less bottlenecked and the DP shifts its attention to a more bottlenecked resource. Shifting of attention also occurs with respect to time, so that the DP can maintain its focus on one resource but shift the focus to different times throughout the planning horizon. All of these shiftings of focus contribute to the construction of a high quality robust plan, in which heavily contended-for resources are planned to be utilized with near maximal efficiency.

7.0 Status

KBLPS is currently being used by the XVIIIth Airborne Corps, Fort Bragg, to plan logistical support for rapid deployment. KBLPS has also been chosen by the US Army Materiel Command as one of the sustainability tools for the "Lousiana Maneuvers". It is also being acquired for use in Europe, Korea and by the Joint Chiefs of Staff J4.

KBLPS is being integrated into the curriculum at the US Command and General Staff College, Fort Leavenworth, Kanas, for training logisticians in distribution planning. It is also being integrated into the Army War College, Carlisle PA, general officers' command post.

8.0 Summary

The Knowledge Based Logistics Planning Shell (KBLPS) is a cutting-edge application combining the best of conventional (X-windows, Motif, etc.) and Artificial Intelligence (Knowledge Base, constraint-directed scheduling algorithms) technologies. There is substantial potential for further growth and development, including adding new applications modules platformed on top of the rich knowledge base already in place. KBLPS represents an emerging capability to accomplish fast and accurate logistics planning at several echelons in the army.

This application is currently undergoing intense evaluation and use by skilled logisticians, supporting their on-going efforts to plan for regional contingencies around the world. The user's, and their senior management's, current enthusiasm and support bode well for further constructive development, extensions, and deployments in the near future.

9.0 Bibliography

[Baykan & Fox, 92] Baykan, C.A., and Fox, M.S., (1992), "WRIGHT: A Constraint Based Spatial Layout System", in *Artificial Intelligence in Engineering Design, Volume I*, Tong, C., and Sriram, D. (Eds.), Orlando Fl: Academic Press.

[Fox et al., 89] M. S. Fox, N. Sadeh, C. Baykan. Constrained Heuristic Search. In Proceedings IJCAI-89, 1989

[Sadeh & Fox, 90] N. Sadeh and M. S. Fox. Variable and Value Ordering Heuristics for Activity-Based Job-Shop Scheduling. In Proceedings of the Fourth International Conference on Expert Systems in Production and Operations Management, 1990.

[Sadeh, 91] N Sadeh. Look-Ahead Techniques for Micro-Opportunistic Job Shop Scheduling, Ph. D. Thesis, CMU, 1991.

[Zweben et al., 90] M. Zweben, M. Dale, R. Gargon. Anytime Rescheduling. In Proceedings of the DARPA Workshop on Innovative Approaches to Planning and Scheduling, 1990.

90C212

A Knowledge-Based System For Military Aircraft Load Planning

Captain James Richardson & Marcel Labbé
Collège militaire royal de Saint-Jean
Richelain, Quebec, J0J 1R0

#149610

Abstract

Load planning of military aircraft is a complex task requiring the expertise of highly trained and experienced personnel. One of the main objectives in the execution of such a task is to achieve the best possible utilization of scarce military airlift resources. Additionally, successful support of combat operations frequently requires responsiveness in modifying conceived load plans (sometimes referred to as "chalks"). The underlying problem consists of loading a wide variety of items into the cargo bay of an aircraft so as to obtain an acceptable aircraft centre of gravity (CG), while giving due consideration to various constraints. To date, only manpower-intensive manual load-planning methods have been employed within the Canadian Forces.

In this paper, we present a knowledge-based system for possible eventual use by military personnel as an aid to produce load plans for various types of transport aircraft used by the Canadian Forces. The developed knowledge base captures, in the form of rules, the domain knowledge employed by experts to arrive at a solution. This tool demonstrates the capabilities offered by combining expert system technology with a powerful graphical user interface in the area of aircraft load planning. A review is conducted of the principal constraints, i.e., loading/offloading and in-flight limitations, to be considered in building a feasible load plan. Finally, a discussion of the modeled loading methodologies is followed by a presentation of the current limitations and possible extensions of the designed prototype.

Key words: *expert system, aircraft cargo loading.*

Introduction

In this paper, we present a knowledge-based expert system, whose purpose is to assist users perform the complex task of load planning for military cargo planes. The requirement for a computer-assisted system capable of improving aircraft utilization and responsiveness in airlift operations within the Canadian Forces (CF) has been recognized for quite some time. To date, however, only manpower-intensive manual load planning methods have been employed.

The notion of applying the expert system (ES) methodology to the aircraft load planning problem has been inspired to a certain extent by successful US Forces initiatives in this area. This initiative was motivated by their requirement for timely and effective aerial delivery of equipment and combat forces to support increasing military commitments abroad. In the same vein, the use of such tools could satisfy the requirement for enhanced strategic mobility of CF elements in a cost-effective manner.

Problem Statement

The need for rapid deployment of a substantial size military force to a distant area of operations can strain scarce airlift resources, thus the importance of effective use of available aircraft. Furthermore, last minute modifications are constantly being made to aircraft load plans once an operation is launched, regardless of how well planned a logistics plan in support of military operations may be. An improved load planning process may thus positively contribute to military readiness which is, in part, a problem of expedient optimization in delivery of fighting troops and materiel to an area of operations.

As summarized by Bayer and Stackwick [2], the problem of interest may be briefly stated as follows: there are various types of aircraft which may be used, each with different capabilities and restrictions and different modes in which it may be loaded, and finally, there is a large variety of equipment to be transported. To compound the problem, the distance to be flown on each mission determines the fuel load, which in turn causes the allowable cargo load and the center of gravity (CG) to change.

Some particularly important aspects of the above stated problem will now be explicitly covered. First addressing the issue of diversity of equipment to be transported, we note that each aircraft can be loaded with cargo including any or all of the following items:

1. wheeled vehicles;
2. palletized cargo;
3. tracked vehicles;
4. helicopters;
5. miscellaneous equipment; and
6. personnel.

The necessity for the loading of a combination of different items may lead to significant compromises. One such compromise pertains to the consideration of aircraft space needed for the placement of passengers and their accompanying baggage in lieu of loading cargo items on some military cargo planes, in which passengers are seated in the cargo compartment next to the equipment. For example, the determination of exact locations of seat hookups to be used to accommodate the required passenger seating capacity has an impact on cargo space available.

Secondly, each mission is characterized in terms of aircraft types to be employed and delivery method. Missions may contain a mix of aircraft types depending upon availability and need. Load arrangements are dependent upon the following delivery methods:

1. strategic air/land;
2. airdrop;
3. low-altitude parachute extraction system (LAPES); and
4. tactical air/land.

For strategic air/land missions, the emphasis is on maximum use of aircraft space and few tactical considerations are involved. For instance, towed loads will be disconnected from their prime mover and telescoped by placing the trailer tongue beneath the rear of the prime mover. In a tactical air/land mission however, speed and ease with which cargo can be on-loaded and offloaded takes precedence, thus towed loads may not be telescoped.

Airdrop operations consist of offloading equipment and personnel by parachute. LAPES is executed when the aircraft is flown very close to the ground over a flat, clear area. In this case, an extraction parachute is deployed from the rear of the aircraft, after which the aircraft reascends. The equipment, mounted on a special platform, is pulled from the aircraft by the drag on the parachute and skids to a halt. Airdrop and LAPES missions must maintain aircraft balance while dropping cargo and must also provide space between items for parachute cables.

Each aircraft type is restricted to a maximum design gross weight (i.e., weight when grounded or upon take-off), a maximum landing gross weight and a maximum zero fuel weight (i.e., weight of aircraft and its load when fuel tanks are empty). The maximum design gross weight includes the fuel load to be carried which is determined, at the outset, on the basis of the distance to be flown and other operational requirements. The established quantity of aircraft fuel in turn influences the allowable load which corresponds to the maximum cargo/personnel weight the aircraft can carry. A load's center-of-balance must fall within a pre-defined CG envelope for all loading methods, otherwise imbalance can render the airplane dynamically unstable in extreme cases. This CG envelope is a distinctive characteristic of each aircraft type and comprises allowable fore and aft CG limits. The challenge is to balance the load fore and aft.

In addition to general spatial considerations related to fitting the maximum cargo into a given space, supplementary constraints pertaining to loading, unloading and in-flight limitations are based on the subdivision of aircraft interiors into regions of uniform constraint values. These constraints may dictate such limitations as the maximum cargo weight capacity, PSI (pounds per square inch) value of tires, axle weight restrictions, cargo height restrictions and maximum weight in pounds per linear foot for any particular constraint region.

The above examined set of constraints is by no means exhaustive. Indeed, a multitude of additional safety and mission related restrictions and parameters such as transport of dangerous goods, cargo/personnel movement priorities, ratio of number of loadmasters to number of passengers per flight, establishment of safety aisles when carrying a mixed configuration of passengers and cargo, limitations pertaining to overseas flights and availability of unobstructed emergency doors. Moreover, other concerns of practical interest, constituting a very general set of rules, may be considered in order to facilitate loading and offloading of materiel.

For instance, in a mixed load configuration, passengers will usually be seated forward of cargo items and palletized cargo will be loaded aft of any rolling stock.

Reconciliation of all constraints and associated lengthy mathematical computations make load planning a very time-consuming, tedious process and initial load plans are not very good from the point of view of load planning objectives. An experienced loadmaster or air cargo specialist can plan a load for one aircraft in 30 minutes to one hour. During deployment, actual loads differ from what was planned, compounding the problem of determining timely aircraft load plans. The response times required for rapid deployment make coping with changes nearly impossible. In such a situation, the time crunch may result in delayed aircraft departures, wasted airlift and decreased support of combat operations.

A mathematical formulation of the aircraft loading problem may be obtained through the application of Operations Research techniques. This methodology has demonstrated serious deficiencies however, since the underlying closely related bin-packing problem (Friesen and Langston [4]) is known to be NP-hard in the strong sense, suggesting it is unlikely that a polynomial solution exists (Garey and Johnson [5]). Indeed, previous approaches in this sphere of activity, for developing an automated tool for planning and executing aircraft loads in a reasonable turnaround time, have proven to be inadequate as they relied on computationally intensive optimization techniques, and lacked the flexibility for modifying either the loading strategy or the many constraints restricting sets of allowable cargo arrangements. On the other hand, as has been pointed out in other NP problems, e.g., the traveling salesman problem, if the requirement of finding the best solution is relaxed to finding a good solution, then heuristic techniques can be applied successfully (Pearl [8]). Consequently, we propose to adopt a heuristic approach to model the procedural knowledge commonly used by experts to load aircraft. In the following section, we undertake a review of two successful endeavors in the area of computer-assisted aircraft load planning.

Successful Applications: AALPS and DMES

In the late 1970s and early 1980s, a fairly extensive effort was made to develop practical solution methods to facilitate the load planning process of military cargo aircraft. Such efforts continue, in fact, today to improve aircraft utilization and facilitate the load planning function. Two US military applications for solving aircraft loading problems will now be presented.

Ignizio [6] reports that efforts to develop the Automated Air Load Planning System (AALPS), a knowledge-based system, were begun in 1978. This expert system uses the expertise of highly trained loadmasters to perform the complicated task of loading US military cargo planes. According to Anderson and Ortiz [1], it takes AALPS a matter of hours for a load planning mission involving 1000 aircraft while an experienced loadmaster normally requires a week to perform the same task.

From a given list of equipment to be transported, AALPS automatically generates cargo loads without further user intervention, which satisfy all constraints while making good use of available space. A convivial user interface is provided using an editor written in C, which maintains a graphical image of aircraft and equipment as items are loaded. These items are treated as active objects, enabling users to select them with a mouse and graphically manipulate them on the aircraft. The system can later revalidate loads altered in this way. In addition to this user interface, AALPS incorporates the following system functional components:

1. an aircraft database which contains detailed aircraft characteristics including dimensions, constraint regions (i.e., division of aircraft interiors into regions of uniform constraint values), and special requirements for each aircraft;
2. an equipment database which contains both dimensional information and special characteristics of cargo items;
3. the loading module which contains procedural knowledge used to build feasible aircraft loads; and
4. a user's options and packages database which contains the user's dynamically updated preferences regarding loading strategies.

The developed system is based on the application of a production system formalism characterized by a search of possible arrangements of cargo items to be loaded. The coded heuristic rules typifying the loading techniques employed by the loadmasters direct the search through the state space. A goal-driven search strategy is implemented in order to identify the appropriate chain of operators—an operator being defined as an action such as sliding an item in the aircraft—which transforms the initial state of the cargo into a valid final arrangement. A set of actions in AALPS is represented as a set of antecedents of a PROLOG goal. The application of each operator is validated through a performance of its associated tests such as two- and three- dimensional spatial checks as well as aircraft-, equipment- and mission-specific checks.

Another effort to automate aircraft load planning, titled the Deployable Mobility Execution System (DMES) was initiated by the Air Force Logistics Management Center (AFLMC) at Gunter Air Force Base, Alabama, in 1981. Cochard and Yost [3] provide the details pertaining to the development, operation and implementation of DMES as an automated tool for cargo loading of military aircraft. They assert that DMES has reduced load planning man-hours by 90 percent and increased airlift utilization by 10 percent.

The developed model uses a modified one-dimensional cutting stock heuristic to generate feasible cargo loads, which the planner can modify using detailed interactive graphics. In the present context, this heuristic consists of "cutting up" the aircraft floor space using templates which represent the objects to be loaded. DMES was written in Pascal for use on stand-alone microcomputers. Ignizio [6] state that DMES has since been converted into C and is now known as CALM (Computer Aided Load Manifest).

Cochard and Yost [3] describe DMES as, simply, a heuristic method for air cargo loading. AALPS on the other hand, is considered an expert system by Anderson and Ortiz [1]. Ignizio [6] qualifies both DMES and AALPS as expert systems since the result in both cases satisfies the definition and purpose of an expert system. As far as the performance of these two systems is concerned, both DMES and AALPS were apparently used with great success during the Grenada rescue operation in 1983 and numerous peacetime field exercises.

In view of the successful research work accomplished in the area of automated aircraft load planning based on the modeling of the loadmaster's knowledge, as described above, we have concentrated our efforts on the development of a knowledge-based expert system. The intent, in doing so, has been to use some valuable concepts applied in AALPS and DMES: especially noteworthy, these include a convivial user interface and aircraft/equipment databases, for incorporation into a prototype ES tailored to the Canadian military environment. The exploration of such an avenue offers the advantage of allowing for the eventual inclusion of an optimization module (or a good heuristic) to efficiently produce a "first-cut" load plan or alternatively, to use the generated load plans as input to a mathematical programming approach for the load planning problem involving many aircraft, e.g., application of goal programming techniques as proposed by Ng [7].

The Envisaged System

A completely automated system, which would not permit any user intervention to rapidly try alternative aircraft load plans, would not be a satisfactory answer to the current problem. The automated tool to be developed should, consequently, be capable of performing the following three basic functions:

1. automatic generation of valid cargo arrangements for each aircraft;
2. generation and validation of user-defined load plans; and
3. user-modification and execution of existing load plans.

The implementation of the new system should be characterized by an improved utilization of military cargo aircraft, efficiency in the verification of constraints and tracking of cargo, computational accuracy and detection/correction of user-caused load planning errors. Valid cargo arrangements should be rapidly generated or updated, i.e., within seconds/minutes rather than minutes/hours, from a given list of cargo and personnel to be transported, with a minimum requirement for user input. The input process would be facilitated through a convivial user interface in the form of appealing computer graphics displaying user options, objects representing the aircraft cargo bay and loaded cargo items, data about aircraft load and other pertinent information. A side-effect benefit of this important feature would be the limited training time required to operate the new system as well as facilitation of the acceptance of the system by personnel in the field to whom it is addressed.

An edit module should be available to permit users to swiftly modify the content of cargo lists by adding/deleting cargo items, or otherwise change information about listed equipment to be deployed. The designed system should make gainful employment of an existing generic equipment database containing both dimensional information and other appropriate descriptive elements of cargo items. Its direct access would allow any user to rapidly obtain standard characteristics of any item type or to tailor this database by adding, deleting or modifying its entries to accommodate local requirements, e.g., fielded version of a standard size truck. In a similar fashion, a preferred loading strategy should be user selectable from a database comprising a set of most commonly used loading methodologies.

The design of a portable tool for operability as a stand-alone system at remote sites is a necessity. In this regard, the utilization of a software package compatible with any commercially available off-the-shelf microcomputer appears to be indicated. It should be possible to print from a terminal all load plans in accordance with the standard forms currently in use within the CF. Load locations are to be exactly defined on the computer generated scale diagrams of the aircraft cargo floor, both laterally and forward and aft, using a set of in-use coordinates.

The system archictecture of the designed prototype shown in Figure 1 allows for the integration of the above discussed features. Two of the three main components of the system will be covered in the next two sections, that is, the expert system and the user interface. These two components communicate with one another through the Windows Dynamic Data Exchange (DDE) feature which is a message protocol for data exchange between Windows applications. From the user's perspective, this feature provides for a transparent access to the expert system via a convivial graphical user interface. The resulting DDE conversation is used to translate the user input into facts and the expert system ouput into graphical form.

The Expert System Component

The loading methodologies, representing the problem solving knowledge commonly applied by highly experienced and trained loadmasters and load planners, have been encoded as production rules using familiar load-planning terms to form the knowledge base of our rule-based expert system. The necessary technical material, e.g., exact locations of passenger seat hookups, was obtained from the appropriate CF publications.[1]

The domain knowledge has been represented using the expert system development shell ART-IM (Automated Reasoning Tool for Information Management). It is a complete toolkit for the development and implementation of rule-based expert systems, using a powerful interactive Windows compatible environment. It supports forward chaining particularly well, which is appropriate for the structure of the aircraft loading problem that dictates that search efficiency is to be achieved with the employment of the data-driven search strategy.

As in AALPS, an initial cargo state contained in a database of facts is iteratively transformed by our system into an acceptable rearrangement by applying a sequence of operators representing the load planner's knowledge. The actions performed by the operators are represented very naturally and appropriately as actions in the right-hand side (rhs) of an ART-IM rule (instead of antecedents of a PROLOG goal as in AALPS). The actions in the rhs of a rule are taken when a rule "fires", and a rule fires when the patterns and conditions of its left-hand side have been satisfied by the facts currently in the ART-IM database. By using such rules, it is possible to model in a natural and straightforward manner the various loading strategies employed by the load planner.

As an example of the modeled loading methodologies, when dealing with palletized cargo, the "60-40" rule of thumb usually applied by the loadmasters in seeking a good first try—one requiring the fewest alterations to achieve a valid load—has been encoded: approximately 60% of the weight is loaded forward of the aircraft CG and the remaining 40% is placed aft of the aircraft CG to counterbalance for the fuel weight whose CG is slightly aft of the aircraft CG. Additionally, the load is built from the aircraft center out toward the ends, placing the heaviest pallet in the center and progressively loading the increasingly lighter objects.

As a general rule, if the aircraft CG of the "first-cut" load plan is not within the optimum range, attempts are first made at sliding the entire load, if possible, so as to reduce/eliminate tail/nose heaviness, otherwise the loaded items are rearranged within the aircraft. The procedure terminates when the CG obtained is within the optimum range, if it exists, or when aircraft stability (i.e., a load plan within acceptable CG limits) has been achieved, depending on the user's preference.

The system can currently perform the load planning task involving either a group of passengers or a mixture of passengers and cargo, such as wheeled vehicles and palletized cargo, for the CC130 Hercules transport aircraft used by the CF.

The User Interface Component

The user interface was implemented using Microsoft Windows 3.1 via Borland C++. The graphical output of the system provides the proposed load plan using terms and forms familiar to the load planner. The aircraft floor plan in the AIR COMMAND CC130 PAYLOAD DISTRIBUTION PLAN (form CF/K 4003) is thus faithfully reproduced in the customary 1" to 60" scale on a standard screen.

---

[1] All consulted CF publications are listed in the bibliography.

The loaded items are all easily identifiable from their graphical representation, drawn to scale and accurately positioned, in the aircraft floor diagram.

Along with the proposed load plan, a list of the loaded items is presented to the load planner in menu form. The items can be clicked on with a mouse to obtain more detailed information and/or to change their location in the load plan, if he so desires. Nested windows pop up predictably in a context-driven manner to reveal information required by the load planner for verifying and/or modifying a load plan. With each modification made to the load plan proposed by the system, the aircraft CG together with other relevant data are automatically recalculated by the system and presented to the load planner in graphical form. Warning messages are issued to signal violations of cargo loading rules and regulations.

Extensions to Current Prototype

Much work remains to be accomplished before the pilot ES reaches the status of a widely accepted full-scale computational aid for day-to-day use by field experts. The following is an outline of some of the areas which require development/refinement as part of any future development efforts.

Efforts in the conception of the developed system have been primarily focused on the loading of the CC130 Hercules aircraft since it is the principal airlift resource used by the CF for the transport of cargo and personnel. The pilot ES may however, be easily adapted to accommodate other existing or future types of CF transport aircraft.

Expansion of the prototype to permit the loading of several aircraft while giving due consideration to the movement priority level assigned to each item to be airlifted would make the ES a very useful planning tool. This would require the addition of formalized knowledge representing the heuristic rules applied by the experts for efficiently selecting cargo from piles of cargo items. In the present context, the notion of piles is a useful concept aiding in the reasoning process and described as conceptual collections of items with common characteristics.

Although the introduced ES has been devised to plan strategic air/land missions, development efforts could be directed toward the inclusion of the capability to support other delivery methods and purely administrative flights. The peculiarity of the latter flights lies in the coverage of several destinations and consequently, the order in which the items are loaded is a prime consideration for facilitating offloading.

It is presently intended to translate the ART-IM written code into a highly comparable, more cost-effective AI language named CLIPS (C Language Integrated Production System).

Conclusion

In this paper, we have addressed the requirement for an automated aircraft load planning system with the capability to produce cargo load plans for the CC130 Hercules aircraft. Successful endeavors in the area of computer-assisted load planning in the military environment were reviewed. A pilot expert system was developed on the basis of features essential to a powerful aircraft load planning tool. Throughout the model building process, particular emphasis has been placed on the construction of a transparent, consistent and complete (with regard to the intended scope of work) knowledge base. Exploiting the advantages of the production system formalism underlying the adopted ES, the inference mechanism has shown to be a reliable and flexible approach in conducting a search of the state space of possible cargo arrangements.

The envisaged system would make DND better equipped to prepare and execute any contingency plan for war and peacetime crisis, while permitting annual savings in using the CF transport aircraft fleet. The devised ES could provide the capability of a full MIS and data exchange interface with other automated systems in developing a coordinated transportation management system.

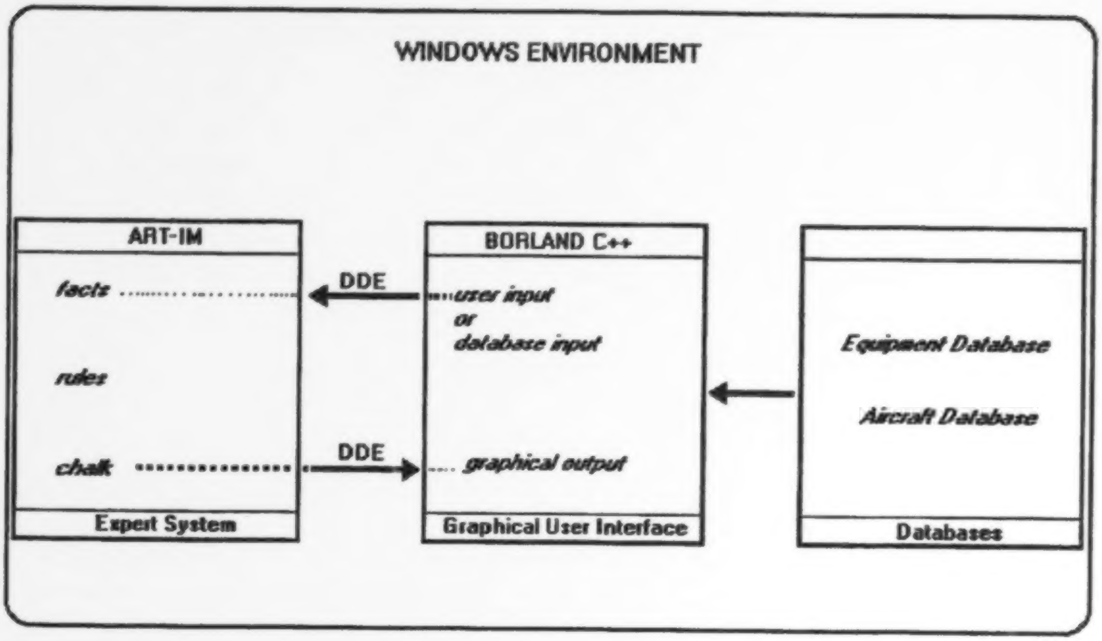

WINDOWS ENVIRONMENT
ART-IM
facts
rules
chalk
Expert System
DDE
DDE
BORLAND C++
user input
or
database input
graphical output
Graphical User Interface
Equipment Database
Aircraft Database
Databases

Figure 1: System architecture

References

[1] Anderson, D., and Ortiz, C., "AALPS: A Knowledge-Based System for Aircraft Loading", *IEEE*, Winter 1987, pp. 71–79.

[2] Bayer, W.K., and Stackwick, D.E., "Airload Planning Made Easy", *Army Logistician*, May-June, 1984, pp. 34–37.

[3] Cochard, D.D., and Yost, K.A., "Improving Utilization of Air Force Cargo Aircraft", *Interfaces*, January-February, 1985, pp. 53–68.

[4] Friesen, D.K., and Langston, M.A., "Bin Packing: On Optimizing the Number of Pieces Packed", *BIT*, 1987, pp. 148–156.

[5] Garey, M.R., and Johnson, D.S., *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W.H. Freeman, San Francisco, California, 1979.

[6] Ignizio, J.P., *Introduction to Expert Systems: The Development and Implementation of Rule-Based Expert Systems*, McGraw-Hill, Houston, Texas, 1991.

[7] Ng, K.Y.K., "A Multicriteria Optimization Approach to Aircraft Loading", *Operations Research*, November-December, 1992, pp. 1200–1205.

[8] Pearl, J., *Heuristics: Intelligent Search Strategies for Computer Problem Solving*, Addison-Wesley, Reading, Massachusets, 1984.

[9] Weight and Balance Data, Canadian Forces Technical Orders (CFTO) C-12-130-000/MW-000, National Defence, 1985.

[10] Manual of Air Movements - Volume 2: Techniques and Equipment, Canadian Forces Air Command Manual (CFACM) 7-400(2), National Defence, 1988.

[11] Manual of Air Movements - Volume 4: Transport Aircraft Loading Procedures, Canadian Forces Air Command Manual (CFACM) 7-400(4), National Defence, 1989.

[12] Air Transport Group CC130 Hercules Configurations/Floor Plans, Canadian Forces Air Command Manual (CFACM) 60-2631, National Defence, 1991.

903412

Potential Applications of Knowledge Based Systems for the Performance Evaluation of Materials in Naval Systems

K.R. Trethewey
Royal Naval Engineering College, Manadon, Plymouth, U.K., PL5 3AQ

P.R. Roberge
Department of Chemistry and Chemical Engineering
Royal Military College, Kingston, Ontario, Canada, K7K 5L0

#147C11

Abstract

The elicitation of knowledge for Knowledge Based Systems (KBS) is a serious problem. The transformation of lifetime industrial expertise into software requires the willing and active participation of domain experts not necessarily inclined or even able to share the details of their thought processes. Knowledge elicitation thus represents a serious difficulty for software developers. Whilst knowledge and experience can be shared by a group of people, expertise is uniquely tinted by the expert's own individuality. This paper discusses the strategies for the formulation of expert knowledge applicable to the people involved in corrosion failures and presents a review of the different areas where the development of KBS would greatly improve the reliability of naval systems, mainly by improving the inhibitive role people can play on the degradation of these systems.

Introduction

Knowledge transfer is rarely a straightforward problem as simple as obtaining the knowledge in some form and encoding it into a computer. For example, experts may be able to perform complex tasks and yet have major problems in understanding and communicating how they do this [1]. These inherent difficulties in the path of developing sound knowledge based system (KBS) have often made their introduction in the workplace precarious and subsequently the accumulation of failures associated with KBS projects has hindered the interest in new development efforts. The interaction between a knowledge engineer and a field expert, in an ideal world, would normally take place at the conceptual stage of a system development. Such an interactive process would then yield the necessary data from which one could gain an understanding of the structure of the factual and reasoning knowledge associated with a relevant domain [2]. In reality knowledge is marked with individuality and while knowledge and experience can be shared by a group of people, expertise is uniquely tinted by the expert's own individuality.

The idea of creating expertise elicitation shells for the development of KBS was born from the need to facilitate an early participation of human experts to such projects. Constructing a prefabricated knowledge structure was thought to be an adequate way to insure and motivate the participation of these experts. By eliciting the targeted domain knowledge with a developmental model of competence, the knowledge engineer would prepare the ground for a win-win situation where both the knowledge engineer and the domain expert would progress rapidly. The criticism of the domain expert, whence given an incomplete but solid model of expertise, should help to reveal obviously missing elements and initiate pertinent improvements. The first step toward the creation of such an elicitation shell should be the identification of a recognised and credible source of expertise that could serve as a design model.

All significant studies into corrosion have reached the same conclusions: that corrosion continues to cost nations dearly, but that most corrosion failures are avoidable [3,4,5]. A review of materials performance maintenance of seawater systems [6] established that corrosion failure is not necessarily a result of bad engineering practice and could be avoided if the efficient initial design was supplemented by a whole life monitoring and control of the environmental and human factors which control systems. Hence the development of KBS would inevitably enhance the probability of effective corrosion management in present and future systems.

The Reliability Factor

When an engineering system functions according to specification, a three-way interaction occurs with complex and variable inputs from people (p), materials (m) and environments (e) [7,8]. The diagnoses of marine corrosion problems can be contained within a hierarchical scheme using a parallel to methods of fault tree analysis [9]. Thus, the consequence or 'top event', a corrosion failure, can be represented by combining these three main contributors into a fault tree (Fig. 1). The contribution to the top event probability (P_{sf}) can then be evaluated with Boolean algebra which leads to Eq. 1 where P_m and P_e are respectively the probability of failure caused by materials and by the environment and $Factor_p$ describes the influence of people on the lifetime of a system. In Eq. 1, $Factor_p$ can be either inhibiting ($Factor_p < 1$) or aggravating ($Factor_p > 1$).

$$P_{sf} = P_m \cdot P_e \cdot Factor_p \qquad (1)$$

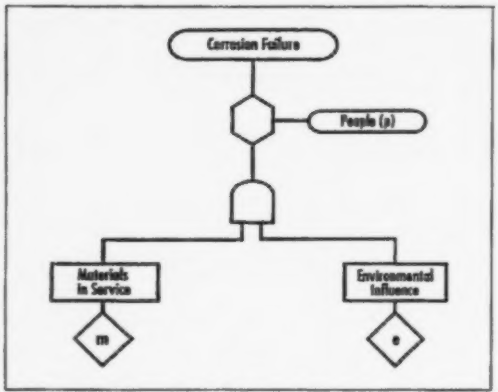


FIGURE 1: Simple fault tree of a corrosion failure.

What might be defined as purely 'mechanical' failures occur when P_m is high and P_e is low. Most well-designed engineering systems in which P_e is approximately 0 achieve good levels of reliability. By reference to an ever-growing database of materials and properties, the designer has a comparatively straightforward task to minimise P_m. Thus, the most successful systems are usually those in which the influence of either the people or the environment is very small and continues to be so throughout the service lifetime. When P_e becomes a significant influence on the increasing of P_{sf}, the incidence of failures by what would then be called 'corrosion' is bound to increase. In stress corrosion cracking (SCC), for example, Parkins [10] has defined a spectrum of behaviour ranging from the almost environment independent transgranular failure to the strongly environment dependent intergranular failure. However, this takes no account of the influence of $Factor_p$. Indeed, it is often changes in $Factor_p$ or P_e during the system lifetime that results in 'unexpected' corrosion failures.

By consideration of the marine industry alone, the evidence for this is overwhelming. Despite the fact that corrosion mechanisms are today understood sufficiently well to enable the specification of efficient corrosion control measures, corrosion problems continue to occur in almost every marine application and represent the single, most important mode of failure of metallic materials. Obviously, there is more to effective corrosion control than mere application of standard methods. Whilst human factors for designers of systems have been expressed in broad terms [11] which take little account of materials or corrosion, this paper focuses more specifically on the implications of $Factor_p$ to reliability in relation to corrosion failure.

The minimising of P_{sf} only through design is extremely difficult to achieve in practice because of the number of ways in which P_m, P_e and $Factor_p$ can vary during the system lifetime. Table I summarises contributions to $Factor_p$, that is, some of the events to which a system is exposed over the whole of its lifetime because of people [12]. Inadequacy at any stage of the sequence will certainly result in a large increase in $Factor_p$ and a consequent increase in P_{sf}, a premature component failure by corrosion. The designer cannot be held responsible when either the manufacturer, for reasons of economy, uses a different material from the one specified, or the user fails to maintain proper control over the environments in which the system operates. Indeed, one of the responsibilities of the Professional Engineer is to control the interactions between people and systems by means of operating and maintenance schedules. Unfortunately, the interactions between people, materials and environments over the full service life are so complex and variable that this laudable aim must be an ideal which in reality is very difficult to achieve. In particular, the marine environment itself is so corrosive and variable that no marine system can be considered 'immune' to corrosion.

The Material Fault Event

The corrosion fault tree in Fig. 1 identifies primary fault events as being either material or environment oriented. Frequently, both components seem inextricably linked and this is a characteristic feature of the complexity of analysing corrosion failures. In naval applications, there is a disappointingly large number of corrosion failures which might be designated 'avoidable'. It is stressed that this is not necessarily the fault of specific ignorance but rather exemplifies the complexity of the effects summarised by Eq. 1. Table II summarises the typical parts of naval marine systems in which corrosion might be expected. The following sections describe some specific materials used in different parts of naval systems.

Main immersed structure

The current choice for hulls in surface ships is mild steel. The over-riding advantage of such a material is its low cost and good weldability by manual and automatic techniques. Corrosion resistance is, however, poor and great reliance is placed upon protective coatings which may be applied at the plate making stage. The major thrust towards cost reduction has been in better fabrication and assembly techniques. It seems unlikely that significantly different materials will be used for conventional surface ships during the next decade.

For submarine service, a stronger steel is required and the quenched and tempered steels [13] are preferred in the Royal Navy. These high-strength steels undergo limited plastic deformation and therefore have an increased likelihood of failure occurring without significant yielding. Good corrosion protection is therefore of increased importance in these high-stress applications if unacceptably high crack growth rates are to be avoided. The steels are also difficult to weld and cause fabrication difficulties in the large sections necessary for submarines. High strength low alloy (HSLA) steels have significant implications for the future development of submarine hulls. They are stronger than current steels which require preheating prior to welding. This preheating imposes difficult working conditions for welders and introduces a considerable cost element to the fabrication program. HSLA steels can be welded with little or no pre-heat, with the result that men can work for longer periods. Protection of hulls for both surface ships and submarines is obviously vital and provided by a combination of both protective coatings and cathodic protection. KBS could be a valuable tool in both instances as will be discussed further below.

Propellers

The precise specification of a propeller for naval service is extremely difficult. Maximum performance, low noise characteristics, corrosion resistance, good castability and the ability to be repaired by welding are all desirable properties. Few materials can meet such a tight set of constraints. Propeller material has evolved from the medium strength manganese bronze alloy through short periods of nickel manganese bronze and Superston to the nickel aluminium bronze in general use today [14]. Its greater tensile strength and wearability have greatly expanded the limits of size and horsepower absorption and have allowed the development of a wide range of versatile propellers and thrusters. Careful control over the microstructure is necessary during casting to avoid the danger of serious selective phase attack experienced in numerous other marine systems. Pitting of propellers or blades can lead to corrosion fatigue failures under the severe operating conditions of naval service [15].

The use of non-ferrous materials for propellers causes interference in other systems. Besides creating an undesirable galvanic influence on the corrosion of the steel hull, it interferes with the operation of the cathodic system, considerably complicating what would otherwise be a comparatively simple procedure. When this kind of interaction occurs, the interfering system can be regarded as part of the environmental influence. This topic will be discussed further in the context of stray currents.

Corrosion of decks and superstructures

Selection of aluminium for decks and parts of superstructures during the 1970s has added to the extent of corrosion of superstructures in the 1980s. In warships, advances in ship technology have resulted in increased topweight which, together with lighter propulsion systems have caused potential stability problems. The value gained in reduced topweight by the use of aluminium was offset by the problems in joining aluminium to steel, first solved in Royal Navy ships by the use of rivets. It was found that vibration in the joint area broke surface

TABLE I: Contributions made by people to the risk of failure of engineering systems: the Factor_p.

Person	Functions
Procurer	Specification of system type
	Inter-system compatibility
	Control of budget
Designer	Geometrical shape and structure
	Choice of materials
	Operational environment
	Use of protective coatings
	Use of electrical protection methods
Manufacturer	Accurate reproduction of design
	Use of specified materials
	Use of correct heat treatments
	Use of correct fabrication techniques
	Correct application of protective coatings
	Monitoring of the environment during storage and transport
User	Correct operating and maintenance procedures
	Use of correct materials to replace failed, worn or corroded parts
	Monitoring of the environment during use
	Maintenance of protective coatings
	Monitoring of electrical protection

TABLE II: Parts of naval marine systems in which corrosion might be expected

PLATFORM

Main immersed structure:
- Inadequate coatings or surface preparation
- Impressed current cathodic protection (ICCP):
  - design problems
  - influence of copper alloy propulsion gear
  - effect of flow and turbulence
  - effect of coatings, fouling

Ancillary Fixtures:
- Bimetallic coupling between propeller and hull
- Electrolytic action from stray currents
- Localised attack:
  - dealloying of copper alloys
  - fatigue cracking under high loading conditions
  - cavitation pits in turbulent areas

DECKS, SUPERSTRUCTURE AND FITTINGS
- General Corrosion
- Corrosion fatigue and fretting in mooring chains and wire rope
- Localised attack caused by bimetallic effects between steel and aluminium

MAIN MACHINERY
- Hot corrosion in gas turbine blades
- Pitting corrosion in gas turbine discs
- Corrosion fatigue in pressure vessels
- Stress corrosion cracking in primary and secondary steam pipe systems
- Pitting attack in secondary steam pipe system

SEAWATER COOLED MACHINERY
- Erosion and impingement in pipework systems
- Multimetallic coupling effects in condensers
- Multimetallic coupling effects in shaft seals
- Main shaft fatigue cracking
- Erosion, pitting, crevice attack and dealloying in pump components and condensers
- Corrosion in desalination plant, membrane degradation

protective coatings and water penetrated the crevices. Serious cracking and material degradation then occurred in the riveted areas in almost all installations. A successful remedy with a proven track record since its introduction in the mid 1980s has been the use of explosion-bonded transition joints [16] used extensively in both Royal and United States Navies. In combination with well-maintained barrier coatings, these aluminium-to-steel joints have provided corrosion-free performance so far. Despite doubts about the value of aluminium in military situations because of its loss of structural stability over 300^{0} C, its use in commercial applications, if not in the Royal Navy, may be of increasing value in the future.

Main Machinery

Interaction of the marine environment with high specification power units originally designed for a different environment was expected to raise many problems. However, the marinisation of aero gas turbine engines for ship applications has largely overcome early problems with hot corrosion. This catastrophic effect on blades caused by presence of sulphur from fuel and ingested sea salts has been tackled with improved materials and advanced coatings on the blades. Such advances are likely to continue.

Probably the greatest risk of corrosion failure in main machinery systems is due to unacceptable rates of crack growth which can occur for many reasons. Changes in thermal stressing, microstructural variations associated with welding stress-corrosion cracking and corrosion fatigue are all superficial terms used to describe the origins of failures which involve a multitude of symbiotic interactions of materials and environments. Fortunately, stress-corrosion cracking is now a well-studied phenomenon and the development of KBS in this area has already been described [17,18]. Cracking problems reported in stainless steel desuperheater tubes [19] are typical of the complexity of the material factor in the reliability equation. The wide ranging effects caused by residual and applied stresses which can be present by the time a component reaches service after a protracted series of manufacturing and installation processes leads to problems which are very difficult to predict and to manage. In nuclear pressure vessels the serious potential danger of corrosion fatigue has been averted by the extremely high levels of care taken in all stages of design, research and development.

Seawater Cooled Machinery

Seawater is used in a variety of auxiliary systems on naval vessels. Besides being used for fighting shipboard fires, it is a coolant for heat exchangers and condensers. There are thus a large array of pumps and pipes associated with the various systems and much scope exists for corrosion problems from a variety of mechanisms. In particular, the presence of electrical coupling through both metallic and electrolytic pathways offers plenty of scope for interaction of 'remote' subsystems. Most of the alloys used for carrying seawater are copper-based as they combine good corrosion resistance and antifouling properties with reasonable cost. However, the extensive history of industrial materials applications and long-established practices of using particular materials in applications for which, today, they are deemed unsuitable by the subject knowledge experts, often results in failures which could be described as 'avoidable'.

The Royal Navy Type 23 frigate has over 17 km of assorted diameter pipework associated with internal plumbing, excluding heat exchangers and condensers [13]. Steel is used for fuel and lubricating oil lines where large diameters are necessary to avoid clogging. Deliberately made accessible, they are easily maintained and replaced when necessary, allowing the use of cheap material. Use of copper pipes is limited to freshwater supplies. Titanium, whilst offering excellent corrosion performance, remains an expensive option and suffers badly from fouling. As a result cupro-nickels are usually preferred for circulating pressurised seawater systems in which 90/10 copper-nickel alloy with 1.5% iron is currently used to reduce susceptibility to stress-corrosion cracking. Relatively high velocities (up to 3.6 m/s) are possible with this alloy. Canadian forces experience with the 90/10 alloy has been generally favourable, but experience in the US Navy has not been so good because of lack of control over flow rate and the poor use of design to minimise turbulent flow [20]. This highlights the importance of the people factor and the need for KBS as a tool at all stages of system life.

Pumps

The corrosion performance of ambient temperature seawater pumps has been generally good and it is only in applications involving different fluids over a wider range of operating conditions that corrosion problems are likely. For example, the use of chlorine at parts per million levels for antifouling purposes is now known to aggravate the corrosion of stainless steels commonly used in pumps. Considerable scope exists for corrosion problems in pumps because of the frequent adoption of multimetallic material combinations, often involving ferrous and copper alloy couples. Common reports of failures in service in the Canadian Navy involve stainless steel (UNS S34700) pump shafts which suffer crevice corrosion, the most severe part of the corrosion not visible with the impeller in place [19]. This kind of problem has resulted in the avoidance of stainless steels by the Canadian Maritime Command and, for this application, more highly alloyed materials such as Monel, Inconel and Incoloy are preferred. Problems which are well-recognised to the experts continue to appear, as in the case of corrosion of G-bronze impellers which occurred in the Canadian Navy some years ago [19]. In 1993 in the Royal Navy Hong Kong Patrol craft, failures occurred in seawater pumps. In this instance, the use of a low performance phosphor bronze for the impellers, presumably on the basis of lower cost, resulted in severe deterioration due to erosion effects. It is not sufficient to blame the engineers directly involved since the wide range of possible materials, environments and conditions, coupled with all the other design and procurement implications, add greatly to the complexity of a corrosion management problem.

Condensers

Condensers have traditionally been a source of corrosion problems with galvanic effects in bimetallic couples between tubes, tubeplates and headers, and erosion corrosion of tubes caused by extreme turbulence. Titanium tubes have largely superseded copper alloy tubes because of the difficulty and expense of retubing, whilst in the Royal Navy copper alloy is still the chosen material for headers. Corrosion problems have not been eliminated in these systems and much effort is currently being expended on application of corrosion management principles to this application [21].

The Environment Fault Event

The role played by the environment in corrosion is particularly complex and needs careful elucidation. Traditionally protection from the environmental factor has been achieved by the use of cathodic protection, inhibitors or barrier coatings. Inhibitors have been very little used in marine systems and will not be discussed further. Together, cathodic protection and paints systems, form the corrosion control protocol for the naval platform.

Cathodic protection

Cathodic protection, the application of sacrificial anodes or impressed currents as a method of protecting steel in seawater, has been extensively studied [22,23] for ships [24] and offshore structures [25,26]. On medium to large size naval ships, impressed current cathodic protection is used in conjunction with protective coatings and, whilst it is generally agreed that this represents the best way of achieving corrosion protection, there is a growing body of evidence that in certain cases the systems installed have been inadequate for the task. Problems which have been identified include: (a) an inability of the system to cope with the asymmetrical coupling of bronze propellers with steel hulls, (b) an inability to react to changes in electrochemistry due to ship motion, (c) a failure of the chosen design to achieve the required potential profile under the variable conditions existing during the life of the system, i.e. coping with paint damage and fouling and (d) unexpected instabilities in multi-zone configurations.

Despite the large amount of published data, there is little data available for the achieved potential profiles of ships under way [27,28]. To date, accepted design methods have been based upon a complex and subjective mixture of theoretical calculations, rules of thumb derived from experience, and more recently, computer simulations based on finite element modelling [29]. It is arguable whether the last are sufficiently well developed to be a reliable method of dealing with the inherent complexities of aqueous corrosion. Control equipment currently supplied for naval use is based upon old designs which are, in general, not capable of providing sufficiently sensitive reaction to the changes occurring in the various operating conditions. There is thus a requirement for the development of new micro-processor controlled technology which can control more complex arrangements of anodes and reference electrodes, sense changing conditions and react accordingly. This field of study offers an ideal opportunity for the development of a KBS which could be used both at the design stage and at the failure analysis level.

Paint schemes

Paint schemes represent the other half of the corrosion protection partnership with cathodic protection. Warships typically have a 250 μm thickness of coal tar epoxy applied to the wetted area of the hull over which the antifouling paint is applied [30]. In theory, the paint acts as a barrier between the hull steel and the aggressive environment, but in practice, the film can be shown to 'wet', allowing the passage of water into pores in the polymeric matrix and completing the circuit for the impressed current cathodic protection. Film performance and breakdown can usually be monitored by the use of electrochemical impedance measurements [31].

Paints have been known to suffer damage by saponification - an effect caused by the build up of hydroxyl ions when the impressed current of the cathodic protection system is too high. Driven largely by the highly demanding operating conditions in the offshore industry, great improvements have been made in coatings technology. As this technology finds increasing use, there is every reason to believe that improvements in corrosion control will continue to be made in all seawater systems, not just for the hull.

Fouling

Fouling, whilst often tackled as part of the corrosion control of the platform, is also a significant aspect of damage to auxiliary seawater systems. Hull fouling is a serious problem. It has been calculated [32] that for a fouled 21,000 d.w.t. (dead weight tonnage) vessel, the average speed loss at full power is two knots and the increase in fuel consumption at 13 knots is 42%. Early optimism about the efficiency of Self-Polishing Co-Polymer (SPC) paint schemes quickly faded with damning evidence of the serious environmental damage caused by the organo-tin toxins used in the formulations. The current situation regarding the legality of use of SPC paints varies considerably from nation to nation. Considerable research is currently under way to find efficient, non-toxic coatings with surfaces which do not attract marine growth. Candidates such as PTFE-based materials, fluorinated polyurethanes and silicone elastomers are presently being evaluated [33]. Fouling in auxiliary systems has been a problem when alloys of metals other than copper (notably titanium) have been used. The problem has most often been tackled by chlorinating, a technique which has brought its own share of new materials failures.

Stray currents

The incidence of electrolytic effects caused by stray currents is one of the most difficult forms of corrosion to diagnose. Often, only circumstantial evidence may be available, for example, in cases where components or systems suffer sudden or unexplained failures. This form of corrosion is a good example of a situation in which the environment is considerably more complex than might have been anticipated. Stray current mechanisms can be rationalised as the interaction of subsystems which, though remote geographically, are electrically connected by the seawater passing through them.

Paradoxically, a particular and growing potential for failures exists with the improvements in barrier coatings for the immersed platforms: as the corrosion protection from the coating improves, the risk of electrolytic action increases. In ships with Glass Reinforced Plastic hulls, for example, there are inevitably a number of metal underwater fixtures such as gratings over intakes for seawater cooling systems, as well as the propeller. In addition, earthing straps are fitted for the ship electrical power system. Besides the usual diesel generators for AC systems, there may also be considerable DC power supplies from batteries. Because of the good conductivity of the sea, it is essential that all the underwater metal components should be in good electrical contact with the earthing straps. Recent instances of serious corrosion of underwater fittings in GRP hulled vessels required a major study to determine whether electrolytic action was the cause or whether a more obvious bimetallic effect was in operation. A well-designed KBS would have been a considerable assistance in this fault analysis.

Conclusion

From this review of the wide range of corrosion problems encountered in marine systems, it is clear that there is no general palliative and that corrosion remains a serious and costly problem to navies. Corrosion problems are essentially interdisciplinary to the currently adopted practices of naval design and procurement. This means that corrosion failures are often not the fault of a specific engineer who may be highly competent in his own area of expertise. The effects of the people factor, coupled with the difficulties in diagnosing, understanding and controlling the complex environmental influences in marine systems, pose extremely difficult corrosion management problems which can be greatly assisted by the kinds of KBS discussed in this paper.

It is believed that long term improvement of the efficiency and cost-effectiveness through control of corrosion will have to be based on methodologies which account for the complex system of interactions between materials, environments and people which is the fundamental cause of most naval corrosion problems. There is an urgent need to incorporate the specific consideration of corrosion factors into the 'whole life' cycle programming and planning, and not merely restrict it to the initial design stages for reasons of short term economy. It must be recognised at all stages in the Procurement and Development program that each change of material, particularly the introduction of alternative materials, and each change of program or performance characteristic can have long term repercussions on the corrosion damage which must be assessed, audited and remedied if truly economic and efficient solutions to these problems are to be achieved.

References

1. B.R. Gaines, "Knowledge Acquisition Systems," in Knowledge Engineering, Vol. 1, Fundamentals, H. Adeli, Ed, McGraw-Hill Pub. Co., New York, 1990, pp. 52-102.
2. E.T. Keravnou and L. Johnson, "Competent Expert System: a Case Study in Fault Diagnosis," McGraw-Hill Pub. Co., 1986, p. 89.
3. T. P. Hoar, "Report of the Committee on Corrosion and Protection," 1971, HMSO.
4. Editorial, Corrosion Prevention and Control, 1980, **27**(3): 1.
5. C. Cabrillac, J. S. L. Leach, P. Marcus and A. Pourbaix, "The Cost of Corrosion in the EEC," Metals and Materials, 1987, **3**(9): 533-536.
6. K.R. Trethewey, "An Overview of Current Naval Marine Corrosion Problems and their Solutions," Proceedings of INEC92, Institute of Marine Engineers, London, U.K., September 1992.
7. R. Wild, "Production and Operations Management - Principles and Techniques," 1984, Holt, Rhineholt and Winston, London, UK.
8. MOD and Dept. of Standardisation, Glasgow, "MOD Practices and Procedures for Reliability and Maintainability," 00-41 (Part 4)/Issue 2: Reliability Engineering, 1989.
9. "Fault Tree Analysis Application Guide," Reliability Analysis Center, Griffiss, NY, U.S.A., 1990.
10. R.N Parkins, "Mechanisms of Stress-Corrosion Cracking," in Corrosion, L.L. Shreir, Ed., Newnes-Butterworths, London, 1976, Vol. I, 8:1-26.
11. MOD and Dept. of Standardisation, Glasgow, "Human Factors for Designers of Equipment," 00-25 (Part 12): Systems, 1989.
12. K. R. Trethewey and J. Chamberlain, "Basic Principles of Corrosion Control," Corrosion for Students of Science and Engineering, 1988, Longman, Harlow, UK.
13. P. G. Hadden, "Materials in a Modern Warship," Metals and Materials, 1988, **4**(7): 412-415.
14. J. A. Norton and J. W. Elliott, "Current Practices and Future Trends in Marine Propeller Design and Manufacture," Marine Technology, 1988, **25**(2 (April)): 118-128.
15. K. R. Trethewey, T. J. Haley and C. C. Clark, "Effect of Ultrasonically-Induced Cavitation on Corrosion Behaviour of a Copper Manganese Aluminium Alloy ," Br. Corr. J., 1988, **23**(1): 55-60.
16. K. R. Trethewey and G. T. Little, "The Use of Explosion-Bonded Transition Joints for the Joining of Aluminium Superstructures to Steel Hulls," Trans. I. Mar. E., 1989, **101**: 57-69.
17. G.D. Bryson and P.R. Roberge, "Development of an empirical model for the Evaluation of Susceptibility to SCC," *Materials Performance: Sulphur and Energy*, P.R. Roberge et al. Eds., Canadian Institute of Metallurgy, Montreal, QC, 1992, pp. 247-257.
18. P.R. Roberge, "Estimating the SCC Intensity of High Strength Aluminum Alloys with Object Oriented Programming," *Light Metals Processing and Applications*, E. Ghali et al. Eds., Canadian Institute of Metallurgy, Montreal, QC, 1993, in press.
19. R. Hollingshead, "Corrosion Problems in the Canadian Maritime Forces," National Defence and Development Branch Technical Memorandum 84/BB, Ottawa, December 1984.
20. N.J. Clayton, "Recent US Navy Experience in Accelerated Deterioration of Copper-Nickel Seawater Piping," 8th. Inter-Naval Corrosion Conference, 1988 Royal Naval Engineering College, Plymouth, U.K., paper 18.

21. K.R. Trethewey and P.R. Roberge "Development of an Expertise Shell for Materials Performance Evaluation in Seawater Systems," Proceedings of NACE/94, National Association of Corrosion Engineers, Houston, TX., 1994, paper # 94378.

22. M. J. Cowling, "Programme Definition Study on Behaviour of Materials in the Ocean Environment," 1990, Scottish Association for Marine Technology.

23. J. H. Morgan, "Cathodic Protection," 1987, NACE, Houston, TX.

24. M. A. Riordan, "A Modern Model for Cathodic Protection Technology," Mat. Perf., 1990, **29**(3): 23-30.

25. D. J. Tighe-Ford, J. N. McGrath and M. P. Wareham, "Evaluation of Warship Impressed Current Cathodic Protection Systems," Trans. I. Mar. E., 1988, **100**: 185-91.

26. R. E. Lye, "A Corrosion Protection System for a North Sea Jacket: Cathodic Protection, Coating Design and Performance," Mat. Perf., 1990, **29**(5): 13-18.

27. R. L. Kean, L. Woolf and M. E. Beardsley, "Corrosion and Protection of Offshore Structures," Metals and Materials, 1990, **6**(12): 792-795.

28. J. N. McGrath, "An Analysis of the Effect of Speed on the Current Demand of the Impressed Current Cathodic Protection System Installed in HMS CHATHAM," 1990, Royal Naval Engineering College, Plymouth, U.K., Research Report RR-90006.

29. M. A. Warne, "Application of Numerical Analysis Techniques," Cathodic Protection Theory and Practice - The Present Status, 1982, Coventry, UK, ICorrST.

30. K. R. Trethewey and J. Chamberlain, "Corrosion for Students of Science and Engineering," 1988, Longman, Harlow, UK, 241.

31. K. R. Trethewey, D. J. Marsh and D. A. Sargeant, "Electrochemical Evaluation of Coal Tar Epoxy Coated Steel in Seawater," Proceedings of 8th. International Congress on Marine Corrosión and Fouling, 1992, Tarranto, Italy.

32. MER, "Fouling-Free Fortunes," Marine Engineers Review, 1991, July, 22.

33. MER, "The Search for an Effective Non-Toxic Antifouling," Marine Engineers Review, 1987, November, 25.

905C20

Range Sensor Array for an Experimental Mars Rover

Larry Korba[1]
National Research Council of Canada
Institute for Information Technology
Ottawa, Ontario K1A 0R6

W. McMath, S. Yeung
Canadian Space Agency
Directorate of Space Mechanics
Ottawa, Ontario K2H 8S2

#149C13

Abstract

Exploration of Mars will be performed using a vehicle or rover operated remotely by a human on Earth. To assist the operator, the Mars rover will be equipped with a cluster of sensors to give feedback about the environment. Sensing the range of objects from the vehicle provides the onboard computer and the operator information that may be used to prevent collisions. Sensors that detect distance based on active infrared techniques are useful for extraterrestrial applications mainly because they do not require an atmosphere in which to operate. The objective of this work was to develop a cost-effective, active, light-based range sensor array for application on an experimental Mars rover. This paper provides a detailed description and some of the characteristics of this new sensor system.

Introduction

A Mars rover exploration program is scheduled for the late 1990's or the late 2000's. The Mars vehicle will support scientific instrumentation used for observation, gathering and analysis of samples from the Martian surface. The rover is expected to be wheeled and equipped with a robot arm. The Canadian Space Agency has been invited by CNES, France to participate in the development of mobile robot navigation systems and onboard manipulators. Developing a system that will aid in understanding the requirements for effective remote teleoperation is the goal of this work. Sensors, predictive control, the sharing of teleoperation and autonomous control are some of the technical challenges in developing the experimental vehicle.

To ease development of systems for this mission, it is important to have an Earth-based experimental vehicle. The experimental vehicle is only the first step in the creation of systems for eventual deployment. For the sake of expediency and to reduce costs, mass market technology has been adapted to meet the requirements of the rover prototype. By taking advantage of large consumer markets, costs for the experimental vehicle are kept to a minimum. For example, the first vehicle prototype is a modified child's sit-upon car. The vision system is an adapted camcorder. The onboard computer is a lap-top computer. Hardware for use in the experimental vehicle is selected if only minor changes are required to have the device do its task effectively.

Operations performed by the rover will be supervised by a human operator. Some tasks will be performed in an autonomous fashion, including automated safety measures for the vehicle and its equipment as a consequence of the significant time delay for communication between vehicle and operator. Information from local range sensors will assist in the operation of the vehicle. For instance, the rover may be commanded to a point one meter away from an object seen in the vehicle's vicinity. It is useful here to have some way of relating the target position to the position of the vehicle in a dynamic way. Sensors that detect the distance of targets from the vehicle are particularly valuable for this purpose. The objective of this work was the development of a low cost array of distance measurement sensors to measure the position of obstacles within an eighty degree field of view toward the front of the vehicle. The system must meet the needs for experimental work, and it would be applicable to the mission vehicle.

Many different types of sensors can detect distance. Ultrasound has been used in robotic navigation for years (1). Unfortunately, ultrasonic range measurement needs a medium for sound propagation. For space applications this is a definite drawback. Electromagnetic techniques (radio waves and light) can measure distance and do not require an atmosphere. Light-based systems can be fairly simple in construction. As with ultrasound, the time it takes for light to travel from a light emitter, reflect off an object and be detected by a sensor may be measured to determine the distance of the object (LIDAR) (2).

---

[1] email address: korba@iit.nrc.ca

Range measurement systems that use LIDAR techniques are often expensive (tens of thousands of dollars). Light however can be used in a variety of ways to detect object distance.

Consumer and commercial markets were investigated to determine whether there is available a sensor that measures distance using a light-based technique that is also inexpensive (several hundred dollars) and requires few modifications to do that function. To this end, the consumer and commercial markets were surveyed (3). Several commercial sensors were tested in search of an appropriate solution. As well, autofocus systems used in cameras were investigated. This investigation included probing the operation of several low cost still cameras and testing of autofocus systems available from Hamamatsu (used in still and video cameras).

Method

The sensor study led to the discovery of a distance measurement sensor that provides a reasonable level of performance and yet each sensor element costs $50.00. The sensor requires a means for attachment to and control by a computer. By means of the computer, the sensors can be operated with simple commands. To apply the sensor effectively, one must understand what the sensor senses. To that end, several tests were performed on the sensor to establish operational characteristics.

Sensor

The sensor adapted for this ranging system is an autofocus module used in the Pentax Zoom 60x camera. Autofocus cameras measure the range of a subject in the camera's field of view. The range information is used to servo the position of the camera's lens from the film to produce a photograph without the subject of interest being blurry. Inexpensive, compact cameras often use focus systems that transduce and convert subject distance into a few digitized levels for the purpose of setting the camera's focus. Although its autofocus module provides a continuous measure of distance over the range of 40 cm to 4000 cm, the Zoom 60x camera uses only a few focus settings.

Fig. 1 is a block diagram of the sensor used in the Zoom 60x autofocus module (4). The output from a light emitting diode (LED) (890 nm) is focussed to project across the field of view of a light transducer. This light transducer is a position sensitive detector (PSD) whose two output currents relate to the position and intensity of the centroid of the light beam falling on it. The light from the LED reflects off a target and is focused by the receiving lens, onto the PSD. As an object moves towards the sensor, its reflected spot moves across the PSD. The distance of the object relates to the position of the reflected spot on the PSD by the expression:

$$L = D^{*}f/x, \qquad (1)$$

where L is the distance to the object, f is the focal length of the receiving lens, D is the distance between the transmitting and receiving lenses, and x is the position of the spot as transduced by the PSD. The module converts the distance x into an output pulse whose width varies between o and 7000 us.

For the Pentax module, these dimensions are: D is 33 mm, f is 18 mm. The size of the PSD's active surface is 2 mm. From the geometry it is clear that small variations in the position or characteristics of any of the components may produce a dramatic change in measured distance.

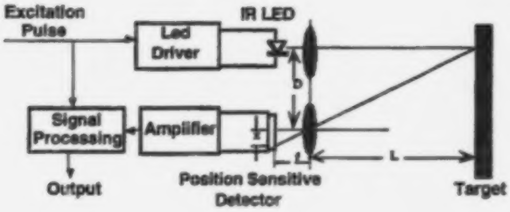


FIGURE 1: Diagram of the autofocus system used in the Pentax 60x camera.

The autofocus module includes a printed circuit board (PCB) that contains the ranging circuit, a light intensity measuring device (for exposure setting), a microcontroller for the camera functions, a power control module and a display device. A separate plastic case (housing the IR LED, PSD and lenses) is attached to the bottom side of the PCB. After investigating the module's operation, a controller was designed to accommodate testing and application of the device as a distance measuring sensor.

Test Descriptions and Results

The axial range response was measured using the arrangement shown in Fig. 2. The sensor was mounted 100 cm above the floor. A black felt backdrop was placed at 450 cm from the sensor to represent a "distant" target (3).

A coloured target of size 30 cm by 30 cm was moved along the principal axis of the sensor to determine the relationship between the sensor output and the position of the target. Targets of two different colours were used, a white matte target and a dark green one. The white target response indicates the effectiveness of the sensor with a good (high reflectance, low specularity) target. The dark green target response indicates the sensor's sensitivity to a poor reflector of the infrared light used by the sensor. This particular colour was chosen through trial and error, using targets of many different colours and measuring the amount of light reflected at the 890 nm wavelength. Comparison between the outputs of the autofocus module to the different targets for any test distance indicates the effect of reflectivity on the sensor's operation. The response of the sensor to these two contrasting targets is shown in Fig. 3. The abscissa indicates the position of the target and the ordinate is the output of the sensor (in microseconds) of the sensor. Thirty-two measurements were taken at each of 16 target position. The target was moved in 30.5 cm (12 in) intervals along the principal axis of the sensor. The difference in response to the two colours becomes apparent at the greater distances (above 2 m.). When this test was repeated in bright sunlight, the test results were unchanged.

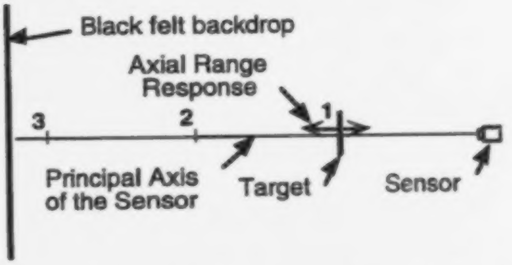


FIGURE 2: Experimental set up for testing the response of the sensor.

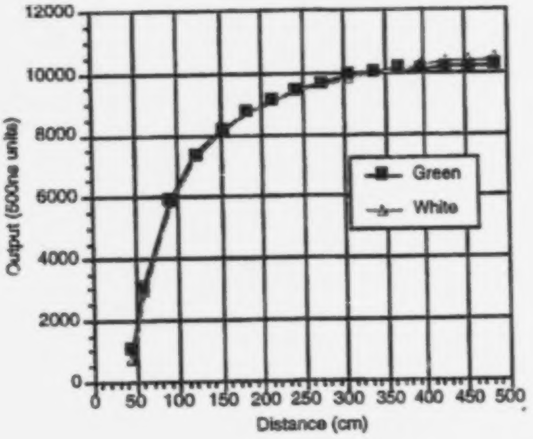


FIGURE 3: Sensor response to targets of good and poor reflectivity.

A measure of the noise in the range output may be determined by examining the standard deviation of multiple sensor readings for the different targets at the 16 different locations along the principal axis of the sensor. The standard deviation for a poor reflector at several meters is a small percentage of the raw sensor output, yet a small variation in the output of the sensor for targets over 200 cm away reflect a large change in output measurement of the sensor due to the geometry of the sensor. To illustrate this, Fig. 4 shows the standard deviation of sensor output as a function of target distance for the two different colours of targets. Note that the error rate for the green target increases precipitously above 2.5 meters.

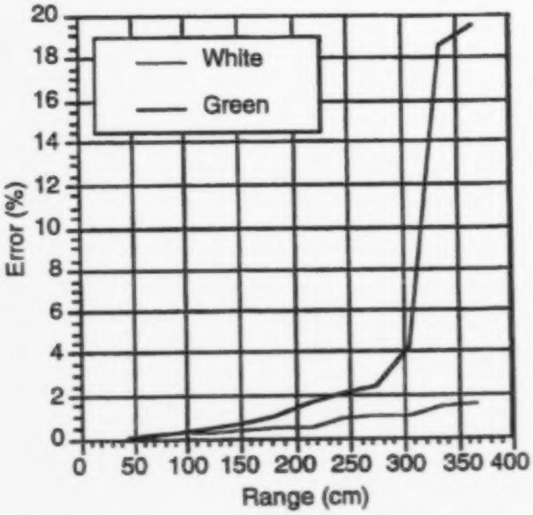


FIGURE 4: Output error as a function of range for the autofocus module range sensor.

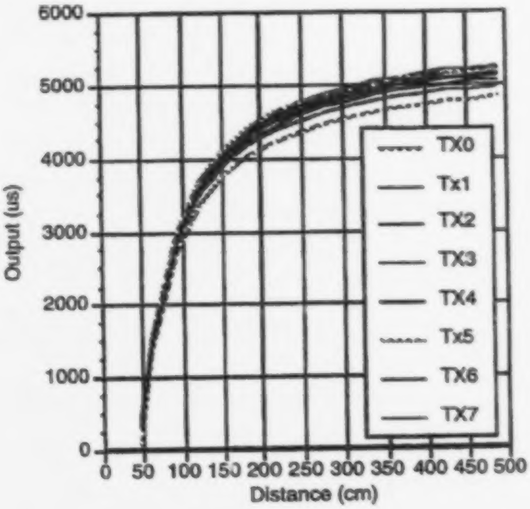


FIGURE 5: Comparison of output response for eight autofocus modules (TX0 to TX7).

A concern when designing a system that may include several of these sensors is the variability of response from sensor to sensor. Fig. 5 shows a family of curves for eight of the range sensors. The variation in output is due to small differences in component characteristics and positions. At a distance of 100 cm the sensor-to-sensor variation is about ±10% error in output range reading for the transduced signal. This error becomes more pronounced with increasing distance. At 250 cm for instance, it is -15% to +30%. Due to this variability, the transducers are calibrated individually and the calibration curves are used to convert the signals from each transducer into output range readings.

There is a compromise between the volume of coverage and measurement accuracy. The projected spot size from the IR LED has an effect on the volume of coverage of the sensor. To reduce the number of sensors required to cover the volume in front of the vehicle, the sensor spot size should be as large as possible. On the other hand, the size and shape of the projected IR spot can distort distance measurement to surfaces that are at an angle to the projected beam or where the spot crosses a change of reflectivity. Poussart et. al. (5) and others (6) describe the phenomena for a laser-based ranging system. An example of the errors associated with spot size is found when attempting to detect the vertical edge of an object when both the IR LED and PSD lie in the horizontal plane. Note that no error occurs when the reflectance transition and IR LED and PSD are in the same plane.

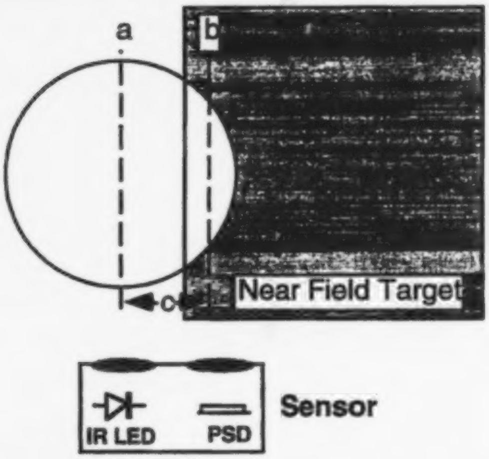


FIGURE 6: This diagram illustrates the effective error produced by a large projected spot size of the IR sensor. Line a is the center axis of projected light point b is the centroid of the reflected light when it moves from a poor reflector (distant object) to a near field target.

Fig. 6 illustrates the problem. As the projected spot moves over the corner of a close object from a more distant object, the centroid of the reflected light intensity moves away from the principal axis of the projected light (from line a to line b). The PSD of the autofocus module transduces the position of the centroid of the intensity of the light spot reflected back from the target. The centroid shift due to the near field target causes a change in output of the PSD. This effect depends on the positioning of the LED side of the detector relative to the reflectance change. For instance, with the arrangement shown in Fig. 6, the output of the range sensor shows an increase in range output as sensor's projected spot passes over the edge of a target. Once the intensity of reflected light from the more distant target is larger than the light from the near target, the output range increases. When viewed with an IR viewing device the size of the spot projected onto a target 200 cm from the sensor is about 10 cm in diameter. As compared to the situation in (5), matters are more complicated since the projected spot has an irregular intensity pattern (depending on the LED and its lens).

To determine the effect of finite spot size on the output of the autofocus module the following arrangement was used. The sensor was mounted horizontally at the center of a precision rotating table. With the sensor pointing outward at the zero degree position of the turntable, a white, matte card was placed, perpendicular to the sensor's principal projection axis. Starting at a position where the projected spot appeared to be bisected by the edge of the card, the angular orientation of the sensor was altered in .5 degree steps in both the clockwise and counterclockwise directions while recording the sensor's output. The test was performed twice, once each for the test card on the left and on the right side of the sensor. The test was repeated for 12 distances ranging from 46 cm to 366 cm.

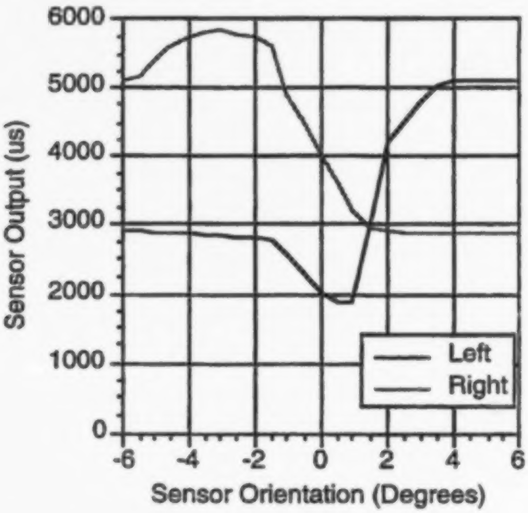


FIGURE 7: Test results for a polar scan over the edge of a near field target at 91 cm. The transducer is oriented with the LED on the left. The zero degree position is where the projected light from the sensor is bisected by the target edge.

Fig. 7 illustrates the results for scanning a target edge. The two traces result from scanning across the edge of a near field target, 91 cm from the sensor, when it is on the left or the right of the zero degree position. This shows the effect of rotating the sensor 180 degrees. When scanning across a range transition that is to the left of the LED, the output of the sensor increases dramatically to a peak of about 30 percent larger in range than the target distance (for any target range). The range output when a target edge is to the right side of the LED produces a range transition that is less steep and the more distant surface appears to be significantly farther away from the sensor than it actually is (over 100% error in range). To minimize this effect, the sensor may be mounted so that the line between the LED and PSD is orthogonal to any expected reflectance transitions or a smaller emitter spot size may be used. For most indoor environments, the effect of vertical target edges on the sensor output can be minimized with the sensor mounted vertically.

When considering number, placement and orientation of range sensors on a vehicle for obstacle avoidance purposes, it is important to know the effective sensitivity volume of the individual sensor. To measure the width of the sensing volume, families of curves for scanning a near field target edge at different distances were analyzed. A 10 percent change in the output of the sensor due to the target was taken to indicate the "edge" of the sensing volume. From this analysis, depending on the target distance, the effective beam width is 10 degrees at 60 cm, 6 degrees at 200 cm and 3 degrees at 300 cm.

Controller

The autofocus module performs all of the functions of the camera. A number of wires interconnect the module and the rest of the camera. The module controls the lens zoom motor, shutter, aperture, film advance and rewind motors, and flash. The ranging system uses the autofocus module as it is operated in the Pentax camera. The camera has a shutter switch with two functions. Full depression of the shutter switch causes the camera to go through a full picture taking cycle. Partial depression of the shutter button allows a user to set the focus point (for the case where the subject will be off to one side of a photograph). By completing depression of the shutter switch, the picture will be taken using the focus setting. Partial depression of the shutter switch closes a switch contact causing the module to measure target range. Distance measurement starts 80 to 90 ms after the focus switch is depressed. The module's output signal is a pulse whose width is related to the target range. To measure range, the autofocus module fires the IR sensor eight times in a row at 12 ms intervals. By averaging the range readings, the autofocus module minimizes the effect of noise from other light sources on the range measurements. The autofocus module requires approximately 250 ms for a complete camera setting cycle.

The microcontroller within the autofocus module monitors the external connections during the camera's operation to ensure that several different elements are functioning correctly. Whenever an error occurs, the module displays an error code on its liquid crystal display, and normal module functioning ceases. If the camera is turned on but none of its functions are exercised during any 2 minute time period, the camera will turn itself off as a power saving features.

To use the autofocus module as a sensor, a method is required to overcome these limitations and perform several other functions. In this design, a microcontroller-based sensor controller performs the following functions:

1) The controller accommodates connection of as many as eight autofocus modules to one controlling computer.

2) It simulates the operation of the camera's servo for all autofocus modules so that they will function normally.

3) It ensures that not one individual autofocus module will turn off its power because of lack of operation.

4) It provides a simple interface so that a computer can easily operate all of the sensors.

5) It sequences the excitation and measurement of any individual sensor, or fire in succession any group of the sensors in a repetitive manner.

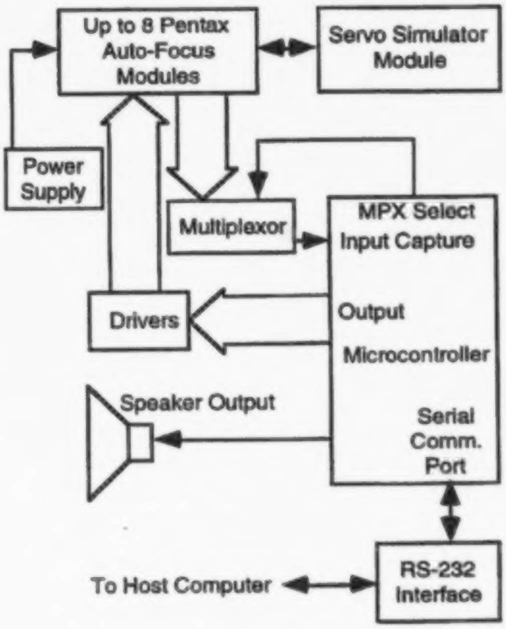


FIGURE 8: Block diagram of the autofocus module sensor system controller.

The block diagram shown in Fig. 8 illustrates the operation of the sensor controller system. The controller operates two switch connections for all of the modules: the power switch and the focus switch. One lens servo simulator circuit produces the appropriate signals for all of the autofocus modules when the sensor is first turned on. The controller manipulates the focus lines of the eight autofocus modules individually. As well, the controller uses two input capture timing channels to measure the output range signal from the modules (7). When exciting more than one sensor in an automatic firing mode, the controller sequences the firing of individual modules so that the arrival of range pulses may be time multiplexed for measurement by the input capture timing channels.

The controller responds to six different types of commands that may be sent to it over an RS-232 communication channel. Commands include: reset the controller, excite any individual sensor to return its range reading, determine the number of sensors connected to the controller, set the individual sensors to be fired in an auto-fire sequence, start or stop an automatic firing sequence wherein selected sensors are fired and their range measurements are taken and returned to the controlling computer, and command the controller to return range data after it is converted through a table into range units. To accomplish this last function, the controller converts the modules' output pulses into range using a conversion table that represents the average of all the sensors (see Fig. 5). This technique provides a very crude way for converting the pulse width from the sensors into range data for testing the operation of the sensors. In normal application, a separate calibration table for each sensor is measured and used by the controlling computer to convert the output of each sensor into a range reading.

The communication protocol interpreted by the controller is based on seven bit ASCII codes. Range data are sixteen bits in length. When using the controller's internal calibration table, the least significant bit of the returned data represents 1 mm of depth. When the raw range data are returned, the least significant bit represents 500 ns in time measurement of the autofocus module's output. Range data are returned in packets of five bytes in length. The data include 1 byte for the transducer number, 3 bytes for the data and 1 byte as a checksum byte. The checksum byte is used to determine whether there may be a data transmission error in the returned data.

Other applications

The system described here was designed for application on an experimental Mars rover vehicle. The sensors were mounted at the front end of the vehicle to provide a eighty degree field of view toward the front of the vehicle. Although the speed of the sensors is adequate for this application (the experimental vehicle will move quite slowly (1 m/min)), a new version of the controller has been developed to provide much higher range sampling rates (at least 250 samples per second). A target application for this system is a scanning range sensor for autonomous vehicle applications. An instrumented rotating mirror may used to change the effective orientation of the field of view of the sensor. Experimental results using a single sensor for this purpose are shown in Fig. 9. In order to minimize the ranging errors due to vertical reflectivity transitions (wall corners), the sensor was mounted vertically (the IR LED above the PSD) at the center of an instrumented turntable. In this plan view of the test area, a dark green filing cabinet in the five o'clock position from the sensor was not detected because of the angular orientation of the object, its distance from the sensor and its colour.

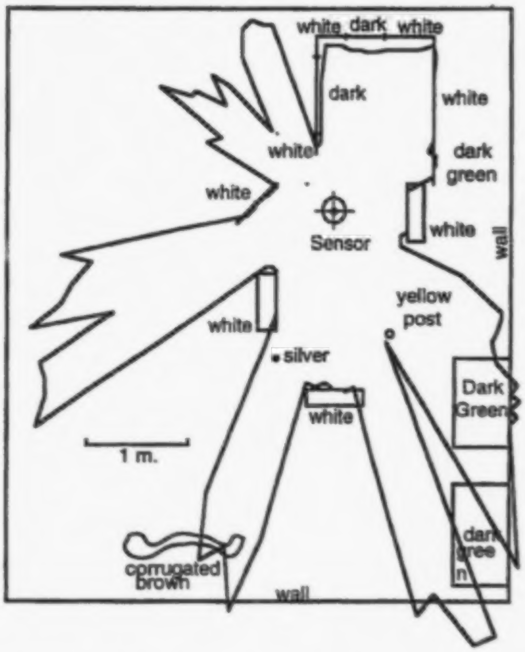


FIGURE 9: Typical results using a scanning sensor to provide a three hundred and sixty field of view around the vehicle. The image line along the lower right wall (4-5 o'clock position) was foreshortened (zig-zag line) for the purposes of this diagram.

Conclusion

The prototype system costs less than $500.00 (Canadian) and provides range sampling of eight sensors in .25 seconds. As witnessed by the sensor characterization, the system measures distance of targets between 40 cm and about 3 m. Range noise produces an error of 1.5 cm at 1 m and 2.5 cm at 3 m for a poor target. Sensor to sensor variation due to mechanical differences requires that each sensor for a bank of sensors is calibrated separately.

An improved sensor controller that provides range readings at over 250 samples per second has been developed. The target application of the device is a scanning range sensor.

Since the sensor uses an IR LED as a projector, it has a large projected spot size compared to laser-based ranging systems. This produces errors due to the movement of the centroid of the reflected light intensity as the spot passes over the transitions in target reflectivity. As witnessed by the data presented earlier, significant range errors will occur if there is a transition in reflectivity in the field of view of the sensor. Depending on the distance from the sensor, the amount of reflectivity change and the orientation of the sensor, errors can be as large as 100% of the true distance. Analysis of this effect is complicated by the fact that the pattern of the projected spot is non-uniform, dependent on the IR LED and its projection lens.

The automatic 35 mm camera market is huge. The economies of scale make the development and manufacture of a good quality range sensor for use in autofocus modules feasible. Practical application of this technology provides robot researchers and industry with an inexpensive, robust ranging system.

Acknowledgments

The authors would like to acknowledge the support of the Canadian Space Agency, Directorate of Space Mechanics.

References

1. Elfes, A., "Sonar-based real-world mapping and navigation," *IEEE Journal of Robotics and Automation*, Vol RA-3, No. 3, pp. 249-265, June 1987.

2. Besl, P.J., "Active, optical imaging sensors," *Machine Vision and Applications*, Springler-Verlag, NY, Vol. 1, pp. 127-152, 1988.

3. L.W. Korba, S. Elgazzar and T. Welch, "Active infrared sensors for robotics," Proceedings of the IMTC '93, IEEE Instrumentation/Measurement Technology Conference, Irvine, CA. May 18-20, 1993. In press.

4. Hamamatsu H2476-01 Evaluation Module Documentation.

5. M. Soucy, D. Laurendeau, D. Poussart and F. Auclair, "Behaviour of the center of gravity of a reflected gaussian laser spot near a surface reflectance discontinuity." *Industrial Metrology 1*, 1990, Elsevier Science Publishers B.V., pp. 261-274.

6. J.P. Donahue, "Intensity and range artifacts in electronic inspection", *Topical meeting on Machine Vision. Technical Digest Series*, Optical Society of America, Vol. 12, pp. 86-89, 1987.

7. *M68HC11 Reference Manual*, M68HC11RM/AD, Motorola, 1991.

905878

Development of a Vision Guided Target Detector and Tracker

Roy Ball & Lionel Pelkowitz
IMAGO Machine Vision Research Inc.
1354 Wellington Street, Ottawa, Ontario, K1Y 3C3

Steve Penzes
Defence Research Establishment Suffield (DRES)
P.O. Box 4000, Medicine Hat, Alberta, T1A 8K6

#149615

Abstract

IMAGO have developed a low cost, extremely portable video target tracker under cost sharing agreements with the Defence Industrial Research Program (DIRP) and the Industrial Research Assistance Program (IRAP). The tracker is a knowledge based system that uses a combination of motion, contrast, size, shape and edge content to initially acquire the target. Tracking uses similar methods together with Kalman filter prediction and logic to coast and automatically reacquire when the target is temporarily obscured. The feedback from the vision system is used to control a pan/tilt head to provide a very stable image of the target. The tracker is available commercially under the IMAGO 100 brand name. A variety of targets have been tracked including rockets, drones, missiles, aircraft, helicopters, towed targets and parachutes.

Several trials using the tracker at the DRES (Defence Research Establishment Suffield) Experimental Proving Ground have been conducted. Successful tracking has been attained under following conditions: clear, overcast and thundercloud skies; long range, extremely poorly resolved, low contrast targets; close range high angular rate targets; rapid aspect changes including contrast reversals; and repeated obscuration of target drone by missile plume. Birds flying through field of view do not cause breaklock.

Introduction

In 1988 IMAGO initiated an internally funded R&D project to develop the IMAGO 100, a PC based video autotracker that would be sufficiently reconfigurable that it could be used for a wide range of applications including:

1. Military target tracking.
2. Perimeter security and electronic sentry.
3. Industrial Robotics.

After the initial definition phase of the project IMAGO were awarded cost sharing grants by IRAP and DIRP.

Features of the IMAGO 100 include: portable 39 lb. pan/tilt head (larger ones also available), low cost, PC-based architecture; fast; accurate; auto-acquisition; cue from radar or other sensor; colour or black and white; multiple TV/FLIR inputs; easily reconfigurable. Multi-tracker systems are available for calculating trajectory and miss distance.

Independently of IMAGO's activities, DRES had identified a requirement for a video autotracker to support both their own research and a number of DND operational trials. Towards the end of 1990 IMAGO and DRES started working together to modify IMAGO's technology to meet DRES requirements.

This paper presents the results of trials conducted at DRES using the IMAGO 100 Video Autotracker and gives a brief summary of some of IMAGO's other activities using the autotracking technology.

Background

The Systems Integration Group (SIG) of the Defence Research Establishment Suffield (DRES) has been involved in R&D into Unmanned Air Vehicle (UAV) technologies since 1979. The prime direction has been in aerial targets and has involved the development of a variety of proof-of-concept systems and flying laboratories. One currently utilized flying laboratory is the High-speed Aeronautical Technologies Testbed (HATT-X). Currently this vehicle uses 19 CRV7 rocket motors which are fired at a rate of two motors per air stage. Flight trials planned for the immediate future will utilize a small turbine motor as a replacement for the rocket motors. A state of the art autonomous autopilot, which is preprogrammed with various manoeuvres and waypoints, provides guidance for the vehicle.

The goal of the R&D program is to enable the technologies required to investigate the use of UAV's as a means to exercise existing and future weapons systems of the Canadian Forces. This goal therefore leads to the requirement for maintaining a level of expertise in the science of weapons systems performance monitoring. Techniques that SIG staff have investigated in this regard are those that use video, telemetry and/or radar systems.

Early in the proof-of-concept (POC) flight testing of ROBOT-X (ROcket BOosted Target - experimental), which was the predecessor to HATT-X, the necessity for video recording of the flight profile was identified. This requirement was satisfied by having SIG and/or contractor personnel operate various camera locations. The operators then recorded the flight of the vehicle by manually tracking with video cameras mounted on fluidic pan/tilt tripods. As the capabilities of the vehicles improved, the flight times became longer and therefore the ranges at which the video operators were required to track became greater.

Various mechanisms were investigated that would allow the operator to track at greater ranges. One of the more promising techniques was the use of video imagery enhancement techniques as a means to provide an automatic video autotracker.

Design Requirements

Client Identification and Requirements

Although the original intent was to develop video trackers for use within the technology based program of SIG, initial definition efforts were to take into account the potential market for such a product both within the government (inter and intra-departmentally) and within the private sector. After reviewing the general types of profile tracking required by the various branches of the Canadian Forces (CF), it was determined that the strongest DND client would be the Army. This was primarily due to the recent acquisition of the ADATS/LLAD (Air Defence Anti Tank System/Low Level Air Defence) and subsequent decision to perform live firing training on the Suffield range. Traditionally the Army had shown an active interest in the use of video/cine technologies for both technical evaluation and training purposes. In addition to the ADATS missile system, the Army has a requirement to track aerial targets during Skyguard gun system trials and during Javelin missile trials.

Autotracker Configuration

After the initial attempt by SIG staff to define an operational configuration, a contractor was located that was capable of performing the R&D work. IMAGO Machine Vision Inc. had a corporate mandate to pursue various aspects of the use of machine vision for the purposes of target tracking. At that time IMAGO was also being partially funded by matching grants from the Canadian government. An initial R&D contract was released to IMAGO in October 1990. In conjunction with a second contract which was released in March 1992, IMAGO was tasked to refine the prototype configuration, acquire the components, integrate the system, perform on site (DRES) testing, and resolve any resultant problems.

After various design iterations were investigated, an interim system configuration was established. The PC provides the feedback signal to control the motion of the pan/tilt head through the use of image matching software and track prediction algorithms. After capturing and digitizing the image from the long focal length lens, the image matching software attempts to locate a target within the field of view (FOV) of the video image. This acquisition is accomplished by adaptively combining motion, contrast, size, shape and edge detection algorithms. Having found the target, the system then initiates a predictive Kalman filter tracking algorithm so as to produce a control signal for the tracking mount. The camera mount then slews to the predicted position and the process is started anew. In the initial stages of the system design, an electronic stabilization system was also included. The purpose of this software subset was to centre the image so that any perceptible motion caused by the tracking mount motion could be negated. The update rate of the control signals to the tracking mount servos was high enough that there was little perceptible motion of the video image even for the 1000mm lens. Therefore the image stabilization software was relegated to a role of secondary importance.

The R&D prototype was ready for field testing in time for the ADATS Perfect Kill training exercise (PK9201) which took place in June 1992. IMAGO then incorporated the observations of both DRES and

PMO LLAD staff in time for the next exercise (PK9301) which took place in February 1993.

Use and Performance

Operations Summary

The operational sequence of events which results in a successful autotrack are straightforward. The operator attempts to manually acquire the target in the FOV by directing the tracking mount with the computer joystick. If acquisition took place with the wide angle lens, the operator roughly centres the image within the FOV and a joystick button is depressed. The tracking software then switches to the long lens camera and initiates an autotrack sequence. Once the system is autotracking the operator can go "hands off" and simply monitor the performance in case intervention is required due to an impending loss of track. The following features and characteristics are included in the software to facilitate this process:

1. The actual image that the autotracking software is monitoring is displayed to the operator. The operator can toggle the displayed image between the two source cameras by pushing a button on the joystick controller. On this image are the superimposed values for the azimuth and elevation readings of the tracking mount in both degrees and mils. This allows the operator to pan and elevate to a given position in the sky as indicated by a third party. During the PK series the information on the position of the target was available to the operators from the ADATS vehicle itself (both EO and radar) and from a DRES tracking radar. This third party cuing proved to be invaluable in the manual visual acquisition process and future plans call for the incorporation of automatic third party cuing.

2. Once autotracking is initiated, the system displays two rectangles. The first is an operator selected, variable sized target position indicator; the second shows the limits of the video search window. The spatial relationship of the two rectangles provides a visual clue as to how well the predictive filter is functioning. As well, the inner rectangle is drawn in various colours depending upon whether the system has a good, weak, or no lock on the target.

3. The VGA display shows some of the values which govern the operation of the system. The software allows the operator to modify these variables during an autotracking sequence. The operator can change the detection threshold, the tracking mount pan/tilt gains, and the Kalman filter smoothing. In addition the operator can preprogram a hot key which toggles the settings to a previously determined set of values.

Results Obtained at DRES

In the time that the R&D prototype has been at DRES for testing, the following types of targets have been engaged by the autotracker:

1. Manned Aircraft - Prior to PK9201 the autotracker was tested against a Piper Seneca II (PA-34-200T) aircraft. This twin engine aircraft was successfully tracked from approximately 2 - 24 km. Although there were some initial problems in the fine tuning of the prediction algorithms, tracking took place at speeds up to 160 kts and at a range of 4 km.

2. Propeller-Driven Subscale Target - This target is approximately 2.5m long with a wingspan of 2.5m and is capable of speeds of 150 kts. Once the autotracker was engaged, the system could track the target consistently between 3 and 12 km. On the outbound leg the system frequently maintained track to almost 20 km. range. Only at the closest point of approach when the target banked sharply to go from an inbound to an outbound leg was the autotracker challenged to maintain lock.

3. During the PK trials, a contractor Lear 35 aircraft would deploy a towed body with a visual augmentation banner. This target would enter a height keeping mode and the tug aircraft speed would be varied up to 200 kts. The towed body length was approximately 4 m and the tow body is about 0.25 m in diameter. Since the majority of the flight profiles presented aspects to the tracker that were either largely inbound or outbound, the frontal area of the target was reduced. In combination with the increased target speed, the reduced target size led to difficulties during the manual acquisition of the target. The autotracker was steady, however, once the target was acquired.

The assessment of the performance of the autotracking algorithms was very positive. Only under known troublesome conditions such as high crossing rate or high rate of change of the target image was performance of the algorithm less than ideal. One circumstance of note was the performance of the autotracker when visual augmentation smoke was used. For some flight profiles the propeller driven UAV was commanded to stream smoke at the farthest end of the inbound leg (12 km). The smoke trail was larger than the target in the FOV but because the tracking algorithm had reached a steady state and because of the regular motion of the target within the FOV, the autotracker was never pulled off of the body of the target by smoke. The biggest shortcoming in the operation of the system occurred when the operator attempted to perform a manual acquisition of a target. The majority of these problems were due to the portion of the software code which governed the manual control of the tracking mount. These problems have been rectified by subsequent engineering modifications to both the joystick hardware and software. Another problem area was in the circumstance of a temporary loss of track such as might be caused by smoke obscuration. This was solved by the introduction of a coast mode. In the event of a loss of track, the tracking mount continues to operate at the last known rates for an operator selectable amount of time. At any time the operator has the option to override the automatic control if it is determined that the target is about to leave the FOV.

Overall the performance of the autotracking system was excellent during the Army trials. The FOV of the telephoto lens in combination with the placement of the cameras relative to the firing ADATS vehicle was such that the system was able to record the flyout of the missile. The video recordings were used by the trials officers for debriefing the gun crews as well as by the engineering officers to troubleshoot system deficiencies. The senior Army staff was convinced that this system could be of long term use as an Army asset and a program task was negotiated in October 1992. The terms of the task were for the purchase and integration of a complete autotracking system as well as for some research work directed to configuration improvement.

Performance Summary

From measurements made at DRES, at Blowpipe camps at Petawawa and in the laboratory, the key performance characteristics of the IMAGO 100 are:

1. Target Minimum Contrast and Size

   Target tracking has been achieved for an unmanned air vehicle target at ranges of 15 to 20 km using a 1000 mm lens. At the extreme range the target is a subclutter point target that sporadically pops up above the clutter.

2. Target Appearance Changes

   Successful unsupervised tracking has been achieved for:

   (a) Very rapid aspect changes
   (b) Extreme brightness variation
   (c) Sun glint
   (d) Rocket plume ignition and extinction

3. Clutter Rejection

   Excellent rejection performance is obtained for the following sources of clutter.

   (a) Clouds.
   (b) Smoke deployed by target even if it is pulsed or alternates black and white.
   (c) Bird fly throughs.
   (d) Very close detonation flashes and smoke clouds.

4. Coast Capability

   Coast and reacquisition is reliable for short obscurations caused by:

   (a) Detonation smoke clouds.
   (b) Missile Plume.
   (c) Clouds.
   (d) Trees or towers. These must not dominate the FOV and the target must have a significant angular velocity to obtain reliable results.

5. Video Tracker Accuracy

   The target off-boresight angle for a point target is typically measured to 1/250 of the field of view. The theoretical limit for real time measurement is 1/500 field of view.

6. Pan/Tilt Unit and Control Accuracy

   (a) Repeatability is 0.1 mrads
   (b) This is degraded by gear backlash and imperfection in the control methodology.
   (c) Measurements on a static target show pointing errors of the order of 0.25 mrads.

7. Tracking Rates

   (a) Maximum pan/tilt slew rate is 90 degrees/second.

   (b) Actual maximum tracking rates depend on:

      (i) target angular acceleration
      (ii) the target flight path

    (iii) the camera FOV

(c) The highest rates will be for a target circling the tracker and will be of the order of 90 degrees/second. The limitation will be the operator's ability to manually acquire the target since he has to approximately match angular rates before engaging auto track.

(d) Highly manoeuvring targets are more difficult to track at high rates.

(e) For lens selection purposes, Table 1 presents extrapolated results for uniform velocity, straight line fly by's.

(f) FASTER RATES are achievable if a trajectory model is available.

8. Target Acquisition

(a) Automatic target detection over the full FOV is standard method.

(b) Can acquire manually with a wider FOV lens and automatically switch to the tracking lens.

(c) Can configure for more precise designation for multi-target scenarios or for severe clutter.

(d) User can select up to 12 waypoints by pointing the tracker manually and selecting CONTROL FUNCTION N. N = 1,...,12. Subsequent single keystroke selection of FUNCTION N causes tracker to slew at high rate to that waypoint.

(e) User can slave tracker to a radar vis RS232 link. IMAGO 100 does the geometric transformations required. User can offset tracker boresight from radar cue point before triggering auto acquisition.

(f) Slave to an encoded binocular will be available in Winter 1993.

Configuration Improvement Plan

The system that was evaluated during the PK series was a DRES laboratory prototype. At this time an R&D contract funded through PMO LLAD tasking has been released to IMAGO to produce a field portable system. The requirement to make this into a field portable system lead to some effort being expended on the design of adequate packaging and system integration. Some minor improvements in the operator interface will be done. Under the terms of the contract, IMAGO is also to perform feasibility studies on the inclusion of automatic third party cuing by radars. In addition, a study on the best method to produce a measure of miss distance through the use of post trial image processing will be done. The next PK trial is tentatively scheduled for February 1994. By that time IMAGO will have delivered a complete single station video autotracker. This will include shockmount racks for the electronic equipment, multiple video cameras, a video playback and editing subsystem, and the next generation tracking mount. In addition, the system will have the next generation software which will be the basis for future research and development work.

The near term work will focus on the integration issues relating to the uses of this system for other scenarios such as anti-aircraft, gun based weapon systems and/or target tracking trials. Preliminary work on the issues relating to the integration of multiple autotrackers is ongoing. The definition phase for the requirements of imagery based post trial processing is also ongoing.

The long term plan calls for the use of multiple autotracking systems working in concert to produce a real time 3D track of UAV's. Such a system would operate automatically once the system has been cued by either a single operator performing a manual acquisition or by a third party radar. Through the use of high framing rate video cameras the terminal flight path of a missile will be recorded as it approaches the target. This will allow the system engineers to assess the endgame of the missile including terminal guidance, miss distance, and warhead performance.

Conclusions

Over the past three years R&D work has been undertaken to produce a capable video autotracking system.

In trials involving live missile firings, the autotracker proved to be invaluable for engineering analysis as well as for operator debriefing because of its capabilities to accurately and reliably track for a wide range of scenarios including very weak targets and the clutter and obscuration phenomena described above.

The experience with the DRES prototype system has proven to be very positive. Based on the expectations of the baseline configuration, only minor modifications have had to be made or are being considered. These shortcomings are currently being addressed; the immediate goal is to package the system as a field-deployable system for the Army.

IMAGO's Other Activities

1. Higher Angular Rates. Although it was not a significant limitation for the DRES requirement, IMAGO have a number of customers including the Canadian Airforce who want to track the missile rather than the target drone. IMAGO have optimised the pan/tilt control to allow tracking at angular rates an order of magnitude faster than was possible at the last DRES trial.
2. Precision Pointing. Naval Research Labs in Washington, D.C. are using the IMAGO 100 to support a number of laser pointing applications. The IMAGO 100 has been carefully calibrated to yield the pointing precision required for laser pointing.
3. Miss Distance. IMAGO are performing trials for the Canadian Army using two IMAGO 100 trackers to calculate target drone/missile miss distance.
4. Electronic Sentry/Perimeter Protection. The IMAGO 300 VIDEO SENTRY is similar to the IMAGO 100 but uses fixed cameras and is optimised for perimeter protection and Sentry operations. Features include zone selection and perspective compensation.

   MPICS is the IMAGO 300 integrated with the PRESEARCH IMAGE CAPTURE SYSTEM. It enhances alarm assessment by quickly showing video images taken before and just after the time of the alarm and highlighting moving objects.
5. Video Based Robotics. IMAGO are conducting research in this area for Automobile Assembly for the Space Station, and for Military applications.

   The research uses the detection and tracking algorithms of the IMAGO 100 augmented with:

   1. Real-time image stabilization.
   2. Stereo imaging.
   3. Single camera model based photogrammetry.

Bibliography

1. Roy Ball, Terry Folinsbee, Tim Pryor, Development of a Vision Guided Robotic Assembly Workcell, DND Workshop on Advanced Technologies in Knowledge Based Systems and Robotics, Nov. 1993.
2. Roy Ball, Terry Folinsbee, Lionel Pelkowitz, Activities in Machine Vision at IMAGO Manufacturing Ltd., 3rd Conference on Military Robotic Applications, Sep. 1991.
3. Terry Folinsbee, et al., Fast Response CCTV Image Recall and Motion Detection, 1993 International Carnahan Conference on Security Technology.
4. S. Penzes, R. Ball, L. Pelkowitz, Development of a UAV Video Autotracker, Proceedings of the 36th Annual Meeting of TTCP Technical Panel HTP-1 (Aerial Targets), July 1993.

TABLE 1

Lens focal Length	Max tracking rate (deg./sec.) Low Clutter	Average Clutter	High Clutter
1600 mm	16	10	4
800	23	16	11
400	32	24	19
200	46	35	28
100	65	51	41
50	large	72	59

Large = approaching 90

Table 1 - Order of magnitude expected maximum tracking rates for straight line, uniform velocity flybys. Use this table for lens selection. These are extrapolated from laboratory measurements and have not been verified in the field. The rates we have encountered in the field have all been less than these.

90C229

ON-LINE ROBOT TRAJECTORY GENERATION THROUGH REAL-TIME OPTICAL TRACKING

Nicholas Krouglicof, Alain Desrochers & Denis Richard
Département de génie de la production automatisée
Université du Québec - École de technologie supérieure
4750, avenue Henri-Julien, Montréal (Québec) H2T 2C8 CANADA
Téléphone : (514) 289-8800 FAX : (514) 289-9042
E-mail : nicholas@gpa.etsmtl.ca

#147C17

Abstract

In general, on-line methods of generating robot trajectories involve manually "jogging" the robot's end-effector along the desired path with the help of a teach pendant or similar device. The joint angles are recorded at a predetermined number of points along the desired trajectory. Due to the limited number of recorded points, the resulting trajectory is typically a piece-wise linear approximation to the desired path. This is adequate for "pick and place" or spray-painting applications, however, many tasks require a significantly more refined trajectory. Such tasks include deburring, machining, grinding and continuous welding.

This paper describes the application of noncontact optical measurement techniques to the on-line generation of robot trajectories. The trajectories are generated in real-time and with a high degree of accuracy by tracking a target probe in all six degrees-of-freedom with a single fixed camera. The problem of determining rigid body pose (position and orientation) from a single perspective view involves the real-time solution of an overdetermined system of nonlinear equations. These equations are based on the Euler-Rodrigues parameters and the quaternion representation of the finite rotation formula. This formulation is computationally efficient and can be implemented in real-time using contemporary Digital Signal Processor (DSP) technology.

The camera, which is based on a Position Sensitive Detector (PSD), was developed at the École de technologie supérieure. The most notable features include high resolution and speed and insensitivity to background illumination. These characteristics were achieved by integrating signal conditioning, data acquisition and processing functions within the camera itself.

The real-time optical tracking techniques discussed in this paper have many applications in the area of robotics. Applications currently under investigation include teleoperation, parameter identification, and real-time error detection and compensation.

Introduction

Industrial robots are currently used for repetitive tasks that generally do not require contact trajectories or force control. In fact, even though robots are flexible machines, their range of application remains limited. This is due to several factors including :

- Operator skill and time required for the programming of even simple tasks (whether it be on-line or off-line).
- The availability of only simple, low-level programming languages which are based solely on positional data (as opposed to high-level, task-oriented languages).
- The failure of conventional robot programming techniques to account for the kinematic and dynamic limitations of the robot (joint overtravel, singularities, linear speed and acceleration).
- Limitations within contemporary robot controllers with respect to the acquisition and integration of process variables (i.e. external sensors).
- Set-up and misalignment problems for repetitive, precision tasks [1,2].

A comprehensive solution to these shortcomings can be achieved through the integration of "intelligent" sensors with a high-level, task-oriented man-machine interface. However, the application of dedicated external position sensors to robotics has been impeded by deficiencies in contemporary sensor technology which make it difficult to monitor process parameters in real-time and with a suitably high degree of accuracy.

This paper describes an innovative optical measurement system which determines position and orientation in six degrees-of-freedom through the interpretation of an active target from a single perspective view. The system, which is capable of accurately tracking an object in real-time, has many potential applications in the field of robotics. For example, the system can be used to automatically generate continuous robot trajectories which describe the position and orientation of the robot's end-effector as a *function of time*. This is essential in applications where the dynamics of the workpiece can not be neglected. When pouring moulds for example, the trajectory of the fluid flow can not be accurately predicted based solely on the position of the robot's end-effector.

Other potential robotic applications for the optical measurement system are in the area of teleoperation, robot parameter identification, and real-time error detection and compensation.

System Description

The four principal modules of the optical tracking system are shown schematically in Fig. 1. The camera and the active target control system, which were developed at the École de technologie supérieure, are autonomous units with the ability to perform certain low-level processing tasks locally. The Digital Signal Processor (DSP) performs the computationally intensive, high-level processing tasks. A conventional PC serves as the man-machine interface and as a supervisory controller for the other modules.

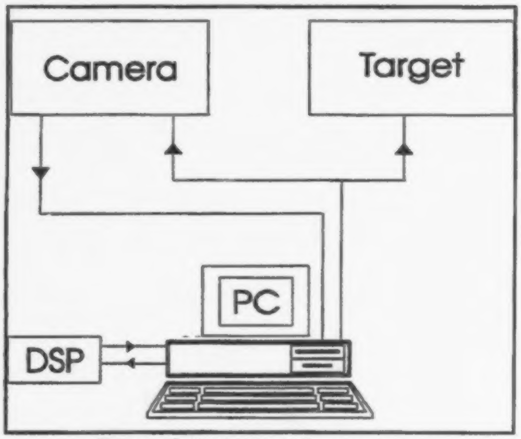


Figure 1 Optical tracking system.

Camera

The camera module developed for the optical tracking system is not a true imaging device in that it does not create an electronic replica of a visual image. The photosensitive element used in this module is a silicon Position-Sensitive Detector (PSD). Reference [2] describes a commercially available measurement system which is based on a Position-Sensitive Detector while reference [3] presents a more detailed analysis of the PSD along with a number of potential applications.

Position-Sensitive Detector (PSD)

The PSD (Fig. 2) is a planar type PIN photodiode with very uniform resistive layers formed over both the top and bottom surfaces. When a light spot is projected onto the sensitive surface of the PSD, electron-hole pairs are generated in the depletion layer directly under the light spot. The electric field in the I-layer causes the electrons and holes to separate, electrons moving to the N-layer, holes to the P-layer. The electrons are then collected by two opposing electrodes located on the resistive surface of the N-layer. The current flowing to each electrode is a function of both the intensity and the X position of the light spot. The holes are collected in a similar manner by two electrodes located on the surface of the P-layer. In this case the current flowing to each electrode is a function of the intensity and Y position of the light spot.

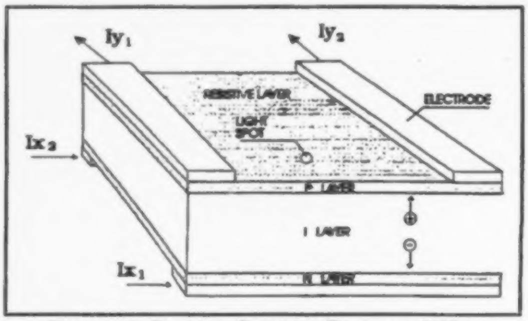


Figure 2: Position-Sensitive Detector (PSD).

Signal processing and acquisition card

The signal processing and acquisition card is shown schematically in Fig. 3. This card, which is integrated within the camera enclosure, has three distinct stages. In the first stage, the low-level current signals provided by the PSD are conditioned by four

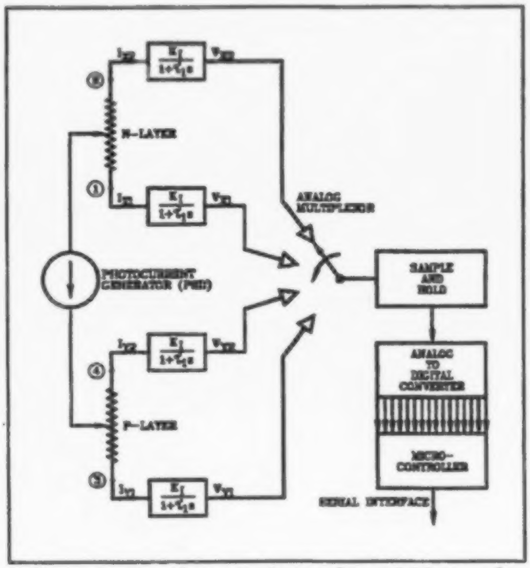


Figure 3: Signal processing and acquisition card.

transimpedance amplifiers. These amplifiers provide a gain of 0.56 X 10^6 V/A and an integration time of approximately 100 μs.

The four voltage signals, V_{X1}, V_{X2}, V_{Y1} and V_{Y2}, provided by the transimpedance amplifiers are channelled through an analog multiplexer and a sample-and-hold circuit (S/H) to a high-speed, 16 bit analog-to-digital converter. The digitized position signals are then sent to a microcontroller which performs some low-level signal processing before transferring the data to the host computer via a high-speed serial interface.

As mentioned above, the four current signals provided by the PSD are dependent on both the intensity and position of the light spot. The two-dimensional coordinates of the light spot, x_d and y_d, can be recovered by normalizing the current signals

with respect to intensity, i.e.:

$$x_d \propto \frac{V_{X1}-V_{X2}}{V_{X1}+V_{X2}}, \quad y_d \propto \frac{V_{Y1}-V_{Y2}}{V_{Y1}+V_{Y2}} \quad (1)$$

This normalization process is performed numerically by either the host computer or by the microcontroller on the signal processing and acquisition card.

Active Target Control System

The resolution of solid-state and scanning-type image sensors is ultimately limited by either the size of the photosensitive elements or the number of scan lines. This is not the case with the optical tracking system described here. Due to the continuous analog nature of the PSD, there is no definable geometric limitations on resolution. The resolution of the PSD is primarily noise limited. The noise that is inherently present due to the discrete nature of the incident light is referred to as photon noise. Photon noise imposes an absolute limit on the resolution that can be attained at a particular light level and for a given integration time regardless of the image sensor.

As with all image sensors, the signal-to-noise ratio can be improved by increasing either the integration time or the intensity of the target light source. A long integration time is impractical in a real-time application, therefore the target light source must have a high radiant power output. In addition, the radiant flux incident on the sensitive surface of the PSD should remain constant irrespective of the position or orientation of the target light source. This is necessary because the magnitude of the radiant flux incident on the detector affects the sensitivity of the PSD to the desired input (i.e. the position of the light spot). Actively controlling the intensity of the light source also prevents saturation as the distance between the camera and the light source decreases.

The active target control system satisfies these criteria by actively controlling the forward current through each of target light sources based on a feedback signal from the host computer. In order to maintain a high signal-to-noise ratio, the target light sources are "pulsed" at a current level considerably higher than the maximum-rated continuous forward current. This is possible providing the duty cycle is limited.

The active target control system is shown schematically in Fig. 4. The target light sources used are gallium aluminum arsenide (GaAlAs) infrared emitting diodes (IRED) with a maximum-rated continuous forward current of 150 mA and a peak forward current of 5 A. The forward current through each IRED is controlled by a power MOSFET (Metal-Oxide-Semiconductor Field-Effect Transistor). The analog controller forces the forward current through each IRED to be proportional to a setpoint value, V_{in}. The setpoint value is generated by a Digital-to-Analog converter (D/A) whereas the feedback value is

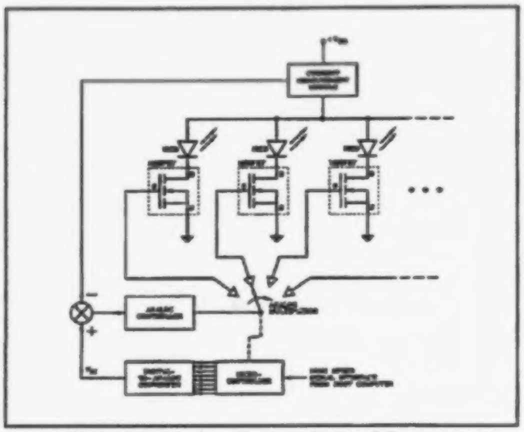


Figure 4: Active Target Control System.

provided by a current measurement module employing a Hall generator principle.

Since only one IRED is active at any one time, the same D/A can be used to control several IREDs through an analog multiplexer. The host computer is continuously calculating the position of each IRED, therefore, it can accurately predict the current required to maintain the intensity at the setpoint value. Basically the required current is inversely proportional to the square of the distance between the IRED and the camera. A new value of V_{in} is computed at each iteration to correct for any discrepancies between the measured and desired intensity values. The appropriate current level for each IRED is transmitted serially from the host computer to the active target control system via an on-board microcontroller.

Low-Level Processing

By incorporating a certain level of intelligence within the camera itself, certain low-level processing tasks can be performed locally prior to the transmission of the data. Such tasks include algorithms for active background cancellation and digital filtering.

Active background cancellation

The two-dimensional current flow through the resistive layer on the surface of the PSD is governed by Laplace's equation. Since Laplace's equation represents a linear, homogeneous, partial differential equation, the superposition principle applies. The current flowing to each electrode when two current sources are applied to the photodiode is equal to the sum of the currents for each source acting independently. Background illumination can effectively be eliminated by sampling the four current signals with the target light source *off* and then subtracting these readings from the current signals sampled with the light source *on*. This simple background cancellation algorithm can be implemented directly on the signal processing and acquisition card.

Digital filtering

Also incorporated within the camera microcontroller software is a median filtering algorithm. This algorithm is applied to each of the four position signals prior to transmission. A median filter operates by classifying the input values in ascending order and selecting the middle or median value. The median filter used in this application is based on five previous sample points.

Host Computer / Digital Signal Processor

As previously stated, the computationally intensive, high-level processing tasks are implemented on a Digital Signal Processor (DSP). The specific device used is a Texas Instruments TMS320C30 DSP operating at 33 MHz. The DSP, which is packaged as a PC half card, communicates with the host computer via a 16-bit bidirectional communications port.

The primary purpose of the DSP is to evaluate rigid body pose - position and orientation - from the image coordinates of a number of body-fixed points (i.e. IREDs). As discussed below, the orientative and range parameters are evaluated iteratively through the application of least-squares analysis on an overdetermined system on nonlinear equations. In general, iterative methods are ill-suited to real-time applications. However, the techniques described in this paper can be implemented using floating-point multiplication and addition alone. Since they do not require floating-point division or the evaluation of a large number of transcendental functions, they are well-suited to contemporary DSP technology.

High-Level Processing

This section of the paper addresses the problem of determining rigid body pose from a noisy, single perspective view of a redundant set of body-fixed points. The noise is filtered through the application of least-squares analysis on an overdetermined system of nonlinear equations. These equations are based on the Euler-Rodrigues parameters and the quaternion representation of the finite rotation formula. A more thorough derivation of these equations is presented in reference 4.

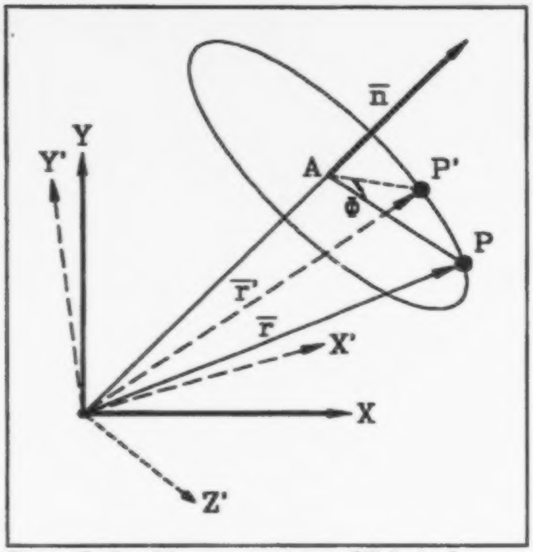


Figure 5: Graphic representation of Euler's theorem.

The Finite Rotation Formula

Euler's theorem states that the most general displacement of a rigid body with a fixed point A is equivalent to a rotation of that body about an axis through A [5]. Let the axis of rotation be defined by the unit vector $\bar{n}$ and the angle of rotation by ϕ. The four Euler-Rodrigues parameters a, b, c and d are defined in terms $\bar{n}$ and ϕ as follows [6]:

$$a = n_x \sin(\frac{\phi}{2}),\quad b = n_y \sin(\frac{\phi}{2}),\quad c = n_z \sin(\frac{\phi}{2}),\quad d = \cos(\frac{\phi}{2}) \tag{2}$$

Note that the parameters a, b and c represent the components of a vector with direction $\bar{n}$ and magnitude $\sin(\phi/2)$ whereas the parameter d represents the scalar quantity $\cos(\phi/2)$.

Consider the two vectors $\bar{r}$ and $\bar{r}'$ which define the position of a body-fixed point *P* before and after a general change in orientation.

$$\begin{aligned} \bar{r} &= x\bar{i} + y\bar{j} + z\bar{k} \\ \bar{r}' &= x'\bar{i} + y'\bar{j} + z'\bar{k} \end{aligned} \tag{3}$$

The finite rotation formula relates the components of the two vectors $\bar{r}$ and $\bar{r}'$ to the Euler-Rodrigues parameters a, b, c and d through a set of four linear, homogeneous equations [7,8].

$$\begin{array}{rrrrr} (x'-x)a + & (y'-y)b + & (z'-z)c & & = 0 \\ & -\ (z'+z)b + & (y'+y)c + & (x'-x)d & = 0 \\ (z'+z)a & & -\ (x'+x)c + & (y'-y)d & = 0 \\ -(y'+y)a + & (x'+x)b + & & +\ (z'-z)d & = 0 \end{array} \tag{4}$$

Note that these equations are not only linear with respect to the Euler-Rodrigues parameters, but also with respect to the coordinates of the body-fixed point before and after a general change in orientation.

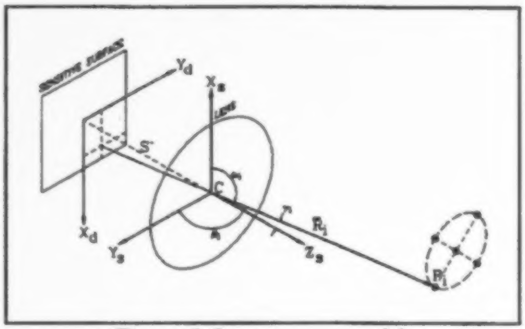


Figure 6: Lens-camera model.

Camera Geometry

Figure 6 represents a conventional lens-camera assembly in which each point in the viewing space is projected towards the focal centre of the lens and onto the sensitive surface of the camera. The position of the image point is defined by two coordinates, x_d and y_d, which are provided by the camera. The vector $\bar{R}_i$ defines the position of target point P_i with respect to the focal centre of the lens. The angles α_i, β_i and γ_i are formed by the vector $\bar{R}_i$ and the X_s, Y_s and Z_s axes respectively. The cosines of these angles are referred to as the direction cosines of $\bar{R}_i$ with respect to the camera coordinate system. The three direction cosines can be expressed as functions of the effective focal length of the lens, S', and the image coordinates, x_d and y_d, as follows:

The vector $\bar{R}_i$ is completely specified by its length, l_i, which is as yet unknown and its direction cosines.

$$\begin{aligned} \cos(\alpha_i) &= x_d \,/\, D \\ \cos(\beta_i) &= y_d \,/\, D \\ \cos(\gamma_i) &= S' \,/\, D \end{aligned} \qquad (5)$$

where: $$D = \sqrt{S'^2 + x_d^2 + y_d^2}$$

$$\bar{R}_i = l_i\left[\cos(\alpha_i)\,\bar{i}_s + \cos(\beta_i)\,\bar{j}_s + \cos(\gamma_i)\,\bar{k}_s\right] \qquad (6)$$

Application of the Finite Rotation Formula

In order to apply the finite rotation formula, one of the target points must be arbitrarily selected to represent the origin of the body-fixed coordinate system (i.e. the fixed point A in Euler's theorem). This point, which remains stationary during a pure change in orientation, is denoted as P_0. For each of the remaining points, the vector $\bar{r}_i'$ defines the *final* position of the point P_i with respect to the origin of the body-fixed coordinate system. The vector $\bar{r}_i'$ can be expressed in terms of the position vectors $\bar{R}_i$ and $\bar{R}_0$ as follows:

$$\bar{r}_i' = \bar{R}_i - \bar{R}_0 \qquad (7)$$

Recall that while the directions of $\bar{R}_i$ and $\bar{R}_0$ are known relative to the camera coordinate system, their lengths, l_i and l_0, have yet to be determined.

Equation 7 provides a means of expressing the final point coordinates, x_i', y_i' and z_i', in terms of the measured direction cosines.

$$\begin{aligned} x_i' &= l_i\cos(\alpha_i) - l_0\cos(\alpha_0) \\ y_i' &= l_i\cos(\beta_i) - l_0\cos(\beta_0) \\ z_i' &= l_i\cos(\gamma_i) - l_0\cos(\gamma_0) \end{aligned} \qquad (8)$$

By substituting equation 8 into equation 4, the quaternion representation of the finite rotation formula can be expressed in terms of the measured direction cosines.

The quaternion representation of the finite rotation formula results in a set of $4(n-1)$ *nonlinear* equations of which only $2(n-1)$ are independent. A two-dimensional perspective projection of a set of n body-fixed points yields $(3+n)$ unknowns; three independent orientative parameters as well as one range parameter for each of the body-fixed points. Thus a minimum of five body-fixed points are required to arrive at a unique solution. A set of six or more body-fixed points represents an overdetermined set of equations which can increase the accuracy of the measurement through the application of least-squares analysis. In order to take full advantage of the redundancy, the orientative and range parameters are evaluated iteratively through the application of nonlinear least-squares estimation.

The Newton-Gauss Method

The Newton-Gauss method for nonlinear least-squares estimation operates in an iterative manner, that is, a vector $[Y]$ is found which satisfies:

$$\begin{aligned} &[J(X^{(k-1)})]^T\,[J(X^{(k-1)})]\,[Y] \\ &\quad = -[J(X^{(k-1)})]^T\,[F(X^{(k-1)})] \end{aligned} \qquad (9)$$

where $[F(X^{(k-1)})]$ represents the vector of function values and $[J(X^{(k-1)})]$ the Jacobian matrix for the system of nonlinear equations under study. Both are evaluated using the most recent approximation to the unknown parameters, $[X^{(k-1)}]$. The new approximations to the unknown parameters are then computed as follows:

$$[X^{(k)}] = [X^{(k-1)}] + [Y] \qquad (10)$$

The procedure continues until the l_2 norm of the vector $[Y]$ is less than a certain threshold ϵ.

The multiplication of the Jacobian matrix by its transpose in equation 9 always results in a symmetric, positive-definite matrix $[A]$ which can be factored into the triangular matrices, $[L]$ and $[L]^T$, by means of Choleski decomposition. The vector $[Y]$ can then be determined by forward followed by backward substitution.

Performance

Convergence of Algorithm

Based on an arbitrary starting point, the pose of the rigid body can generally be determined within six iterations. If the velocity of the body is small, and if it is being tracked continuously, then the initial estimates of the unknown parameters are typically accurate and significantly fewer iterations are required. Table 1 shows the rotational and range data after each iteration for a typical trial run.

In this particular example, five body-fixed points are included in the regression model. Points 1 through 4 lie equally spaced on a 0.3 m circle. Point 0 defines the centre of the circle. The target pattern as seen by the camera is shown schematically in Fig. 7. The four Euler-Rodrigues parameters, a, b, c and d, are initially set to 0,0,0,1 while the range parameter for each point is arbitrarily set to 2 meters. Using only the two-dimensional image coordinates of the target points, the algorithm computes the orientation and range parameters to within five significant digits after five iterations.

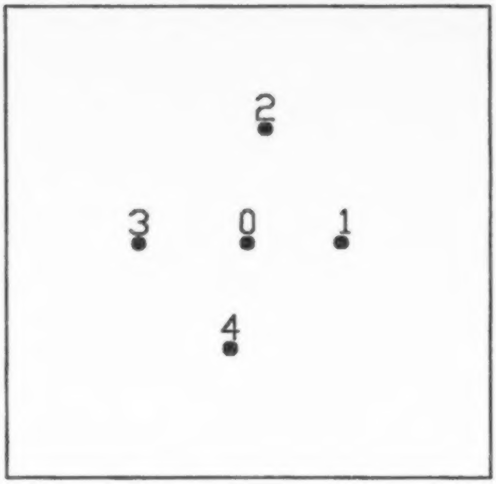


Figure 8: Sample target as seen by the camera.

TABLE I: Rotational and Range Data

ITER	YAW	ROLL	PITCH	RANGE
0	0.000	0.000	0.000	2.00000
1	3.245	1.646	1.463	1.08341
2	11.965	21.875	37.443	1.07782
3	10.108	19.914	30.869	1.00297
4	9.999	19.994	30.006	1.00005
5	10.000	20.000	30.000	1.00000

Resolution

As mentioned previously, the PSD is purely an analog device with is no definable geometric limitation on resolution. Consequently, the resolution of the PSD is primarily noise limited.

The system described in this paper achieves a resolution corresponding to approximately 11 noise-free bits. This is demonstrated graphically in Fig. 8 which shows the distribution of the coordinate data provided by the camera for thirty acquisitions of the same stationary target point. The time required for each acquisition is approximately 1.2 ms.

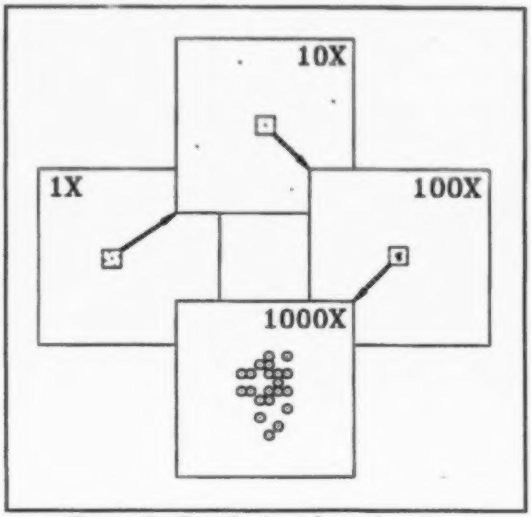


Figure 7:: Distribution of sample points.

Calibration

Most commercially available PSD-type photodiodes exhibit poor linearity particularly towards the extremities of the sensitive surface. In contrast to this, the PSD used in this study is highly linear and provides accurate results based on a simple straight-line input-output relationship. Nevertheless, a two-dimensional calibration table was introduced to compensate for the nonlinearities in the PSD as well as for the lens distortion. The calibration table was determined by accurately displacing a target light source by means of a Computer-Numerically Control (CNC) machine tool.

Conclusions

This paper describes the development of noncontact optical measurement techniques for the on-line generation of robot trajectories. The system, which combines high accuracy and real-time performance, is capable of tracking a target probe in all six degrees-of-freedom with a single fixed camera. The orientative and range parameters are evaluated iteratively through the application of least-squares analysis on an overdetermined system of nonlinear equations.

In general, iterative methods are ill-suited to real-time applications. However, the techniques described in this paper can be implemented using floating point multiplication and addition alone. Since they do not require floating-point division or the evaluation of transcendental functions, they can generally be implemented in real-time using contemporary Digital Signal Processor (DSP) technology. For example, each iteration of the Newton-Gauss method requires approximately 1 ms on a Texas Instruments TMS320C30 DSP.

The method is very robust; the solution does not degenerate for any angular orientation and a complete solution can generally be obtained within 6 iterations from an arbitrary starting point.

Acknowledgments

This work was supported by a PSIR grant (Programme de Support Institutionnel de la Recherche) from the École de technologie supérieure as well as research grants from the Natural Sciences and Engineering Research Council (NSERC).

The authors would like to express their gratitude to Mr. Daniel Faucher and Mr. Jocelyn Lalande for their valuable contribution to this project.

References

1. H.W. Stone, A.C. Sanderson, "Arm Signature Identification", *The Robotics Institute 1986 Annual Research Review*, pp. 21-31, 1986.

2. E.K. Antonsson, R.W. Mann, "Automatic 6-D.O.F. Kinematic Trajectory Acquisition and Analysis", *Journal of Dynamic Systems, Measurement, and Control*, vol. 111 pp. 31-39, March 1986.

3. N. Krouglicof, "Noncontact Position and Orientation Measurement Techniques for Real-Time Systems", *Ph.D. Thesis, Department of Mechanical Engineering, Concordia University*, February 1991.

4. N. Krouglicof, "Rigid-Body Pose Measurement from a Single Perspective View", *Proceeding of the International Conference on Intelligent Autonomous Systems: IAS-3*, pp. 368-377, February 1993.

5. L. Euler, "Formulae Generales pro Translatione Quacunque Corporum Rigidorum", *Novi Commentari Acad. Imp. Petrop.*, vol. 20, pp.189-207, 1775.

6. H. Cheng, and K.C. Gupta, "An Historical Note on Finite Rotations", *ASME Journal of Applied Mechanics*, vol. 56, pp. 139-145, March 1989.

7. G.H. Schut, "On Exact Linear Equations for the Computation of the Rotational Elements of Absolute Orientation", *Photogrammetria*, vol. 17, No. 1, pp. 34-37 , 1960.

8. G.H. Schut, *Formation of Strips from Independent Models*, AP-PR36, NRC-9695, Division of Physics, National Research Council of Canada, Ottawa, July 1967.

905215

Tactile and Proximity Sensor System

#149618

Normand Paquin, Jeffrey Blezius, Daphne Sfeir
MPB Technologies, Inc., 151 Hymus Blvd.,
Pointe Claire, Qc., H9R 1E9.

Peter Wojcik, Keith Chrystall
Alberta Research Council, 3rd Floor, 6815 - 8th Street NE
Calgary, Alberta, T2E 7H7

Abstract

Advances in remote telerobotic operation are required for robots to be used efficiently in hazardous environments such as space, nuclear, or underwater domains. Such advances must include enhanced manipulative capabilities of robotic manipulators, as well as the ability to operate robots at large distances. A tactile and proximity sensor system concept, conceived for potential application on the Evolutionary Mobile Servicing System on Space Station Freedom is reviewed. In particular, sensor considerations in this environment, the architecture of the proposed sensor and control system, remote teleoperation, and the expected benefits with respect to the enhancement of dexterous operational capabilities as defined from strawman tasks for the manipulator are discussed.

Introduction

The usefulness of robot manipulators can be significantly increased if their ability to perceive their environment is improved. A purely visual approach to sensing has its limitations. Even in telerobotic operations with no autonomy, proximity sensors are desirable for the final approach to grasping an object when a visual sensor would be blocked by the robot gripper. Tactile sensors can be utilized to position the robot gripper for a good grasp, and to ensure a secure grasp without damage. The advantages of proximity and tactile sensor systems become much greater in semi-autonomous operations in which tactile and proximity sensor data can be utilized by the robot to complete operations under local feedback control with input from the operator requiring only higher level commands.

In order to develop a concept for an effective tactile and proximity sensor system (TPS), strawman tasks emphasizing the capability of the TPS system were identified. These tasks focused on the enhancements of the manipulation capabilities of the Mobile Servicing System (MSS) being developed for Space Station Freedom and therefore have the potential for application to the Evolutionary Mobile Servicing System (EMSS). The two strawman tasks that were retained and subsequently decomposed into subtasks are (i) the unmating/mating of a connector, and (ii) the grasping of an unknown object. These tasks focus on the enhancement of manipulation capabilities for the Special Purpose Dextrous Manipulator (SPDM), a subsystem of the MSS.

Strawman Tasks

Fundamentally the SPDM is to grasp an object and manipulate it in some way. Vision will be useful in the initial approach to an object. Close-in, vision systems could be obstructed and the robotic system would require tactile and proximity information. A tactile and proximity sensor system can also be utilized to provide basic information such as whether the approach to an object is correct and whether the object has been grasped correctly. In fact, one of the advantages of such a TPS system is that the information gathered can be used at the local level to control various aspects of object manipulation such as the approach speed, angle of approach, forces applied, etc..., relieving the operator from these potentially exhausting and difficult fine manipulation tasks in teleoperation mode. This has important implications for remote teleoperation when time delays for control and sensor signals are significant. In a related vein, a tactile and proximity sensor system will greatly reduce any dependence on optical target systems which are subject to shadows, occlusions, and variations in illumination.

Strawman task #1: unmating/mating a connector

The first strawman task retained is the unmating/mating of a connector. A bayonet type of connector serves as an example. The connector is assumed to be on an Orbital Replacement Unit (ORU) which is about to be replaced. To unmate the connector, the outer ring must be pushed inwards, turned 90° and pulled back past the retaining pin. The plug can then be pulled out. The plug is then to be connected either directly to the ORU replacement or to a temporary connection point - it is considered undesirable that connectors be allowed to float freely.

The ideal end goal to this strawman task would be to reach the point where an operator can identify a connector and issue the simple command "unmate the connector". The robot would autonomously identify the type of connector, establish a task plan, approach it, grasp it, and unmate it and be ready for the next task. This goal may not be too far off in the future. Each task or subtask can have a varying degree of autonomy. As teleoperation capabilities evolve towards full autonomy, each subtask can move towards that direction.

In order to determine how a tactile and proximity sensor would best serve in increasing the autonomy of these operations and allow the SPDM to complete these more effectively, the strawman task was broken down into subtasks, with emphasis on what proximity or tactile information would be useful. The following is a brief summary of the task breakdown. For each subtask, an analysis was conducted in order to determine what type of information would be required, specifications for this information, and what local sensor-based control loops should be incorporated within the TPS. An example follows.

. The task begins with the SPDM within reach of the connector demating and mating points and in view of the connector. In sequence, the identified subtasks for the unmating segment of the task are: (i) location of the connector (world model or vision based); (ii) move the robot hand to a close-in position (20-30 cm from the connector); (iii) move the robot hand from the close-in position to the pre-grasp position (<20-30mm); (iv) align for grasping; (vi) move robot hand to ring grasping position; (vii) grasp with compliance; (viii) grasp assessment; (ix) unlocking of the connector; (x) move connector away from ORU. For the mating segment, new subtasks included (a) insertion with compliance and (b) insertion alignment.

Looking in more detail at the "Align for grasping" subtask, the following analysis can be made. Firstly, it is expected that the visual-assisted alignment is relatively good. However, this close-in (<30mm), a view of the connector could well be obstructed. Proximity sensors could be used to provide fine correction in positioning and alignment. For example, a simultaneous measurement of the distance between the connector surface and four proximity sensors on the robot hand could be processed to give the relative position of the connector with respect to the hand. Proper alignment will assist in maximizing the smoothness of the grasp. The complexity of the algorithms needed to complete this task will be directly related to the complexity of the connector. These fine approach maneuvers should be done under the control of the TPS: the operator would therefore not have to interpret complex data (relative position, angle, rotational orientation, etc...). The final component of the subtask would be to open the robot hand sufficiently to grasp the connector. If the connector is known, this could be done autonomously. By considering the capabilities of the SPDM such as its maximum speed, force control limits, acceleration limits, the analysis can be advanced one step further to determine requirements for the TPS in terms of the accuracy of the data it should supply and the data refresher rate that should be achieved.

Strawman task #2: Grasp unknown object

Two different situations where grasping an unknown object can be expected to arise. As a first case, the robot hand might be required to grasp a small object. The second is that the SPDM may have to grasp a larger object with small features which should allow for a grasp to take place. The object could be an ORU that is to be grasped from an unplanned approach, a damaged ORU whose grapple lug is either damaged or inaccessible, or an unknown object that is not part of the original data base.

As for the first strawman task, this task was broken up into a series of subtasks. In sequence, these were (i) location of potential grasp points; (ii) mapping of potential grasp points; (iii) move robot hand to pre-grasp position (20-30 mm); (iv) move robot hand to grasp position; (v) grasp with compliance; (vi) assess object compliance; (vii) secure grasp; and (viii) grasp assessment. These were also analysed as described for the subtask of strawman task #1.

The analysis of the tasks were converted into specifications for the tactile and proximity sensor system. Table I provides a summary of the processing capabilities that are to be provided by the TPS based on the strawman tasks analyzed. The next step is to examine how this TPS should be implemented, starting with the selection of sensors to the architecture that should be developed.

TABLE I: TACTILE AND PROXIMITY SENSOR SYSTEM: CAPABILITIES.

Performance/Parameter	TPS goals
Data to be gathered and processed:	
Proximity:	Relative distance between object and nine proximity sensors, angle with respect to a flat surface.
Tactile:	Taxel image from two arrays, absolute force, moments at fingertips, slip status.
Sensor-based control loops	Control under Real-time operating system.
	Discrete Event Dynamic System. This software permits transfer from one set of sensors to another and from the operator to local sensor based control loops and vice versa. Implies sensor group selectivity capability.
Specific control loops: For all these control loops, the TPS will autonomously gather, condition, and interpret the data. The system will allow sensor-based closure of local control loops by sending appropriate commands to the manipulator arm and hand controllers. The loop will be closed with human like reaction times. At the same time, information previously selected by the operator will be displayed on a screen.	Gripper approach to object (especially speed). Align. Grasping with compliance. Grasp Assessment. Unlock/lock control loop Insert with compliance Mapping of a work surface Haptic hardness sensing.

Sensor Selection

It is necessary to evaluate the sensor technology most suitable for the TPS. The analysis was completed for both proximity and tactile sensors. For proximity sensors, our evaluation based on the criteria given below, favored capacitance-based and laser-based triangulation proximity sensors. The analysis of tactile sensor technology was more difficult as most tactile sensors are still at the laboratory prototype stage. Information that can be gathered on various types of sensors is limited and discriminating between the various techniques is a difficult task. The Analytic Hierarchy Process, ideally suited for an evaluation based on dissimilar criteria and unknown factors, was applied to the tactile sensor evaluation. The results of this analysis favored capacitance based technology followed by piezoresitive and piezoelectric techniques.

Evaluation Criteria

An extensive review of applicable criteria was applied to the evaluation. These were grouped into subgroups as given below.

The first subgroup is focused on low earth orbit (LEO) endurance. For space application, the outer surface of the sensor must be either inherently resistant to attack by atomic oxygen (i.e. composed of resistant materials) or be coatable by a suitable protective material. The sensor must be able to operate under exposure to the extreme temperatures. The range of the temperature for the sensors, based on the assumption that the sensors would be in contact with objects attached to the space station is narrower than the extremes of the LEO [Dr. V. Wehrle, Canadian Space Agency, private communication]. The outer surface of the sensor must also be able to survive exposure to the long-term vacuum of LEO. Because of the deep penetration of ionizing radiation, the complete sensor should be able to survive the dose of ionizing radiation that will be experienced on the space station or be designed such that it can be protected for extensive periods of time. The outer surface of the sensor should not be degraded by UV radiation.

During the life of the sensor in LEO, its surface will be breached by collisions with micrometeoroids, creating craters. It will likely have a crater as large as 0.4 mm after the 30-year flight, and will have about 5 20-μm craters per cm^2 after just 1 year of LEO exposure. These estimates are based on continuous full exposure to the LEO and statistical probabilities. These estimates would change significantly if the sensors are shielded and/or protected when the MSS is not in operation. The sensor should not suffer catastrophic (complete) failure due to such an event. A small reduction in the performance of the sensor would be acceptable. For proximity sensors, an additional environmental criteria is optical interference susceptibility.

The second subgroup falls under calibration requirements. These should be minimized. Sensor and electronic drift, initialization needs, sensor uniformity are in this subgroup.

The third subgroup dealt with sensor ruggedness. Because replacement and maintenance activities in orbit are expensive, it is desirable that the sensor be rugged. This group includes reliability, durability, overload survival capacity, moving parts requirements, flexibility (impact resistance/force attenuation, compliance, repetitive strain dissipation), and maintenance needs.

The fourth group focused on sensor characteristics. This encompassed whether the output of the sensor could be applied to proximity or tactile sensing, whether slip detection was achievable, whether absolute or relative readings could be obtained, and if so, whether these would be binary or continuous in nature. Other criteria in this group included adjustability, applicability, linearity, hysteresis, dynamic range, data processing requirements, and response time. An important subset of the sensor properties is related to their signal to noise performance. This formed a fifth criteria subgroup including resolution (distance, force, spatial, angular, cross-talk, and dynamic stability as applicable to each sensor type, namely proximity and tactile sensors).

The sixth criteria subgroup was concerned with physical characteristics. The group included physical dimensions, flexibility (adaptivity to different fingers, i.e. curved surfaces), mass, power needs and the number of connections required. The TPS, as for all space subsystems, must minimize all these factors.

The final criteria subgroup was based on economic considerations. The state of development of the sensors were considered as well as potential manufacturing costs, potential markets and competitors (commercialization potential), and the breadth of potential applications of the sensor technology.

Analytic Hierarchy Process

The Analytic Hierarchy Process permits the introduction of individual pairwise comparative judgements to be made to resolve conflicts that normally arise when incomparable criteria are the basis of the selection process. The Analytic hierarchy process has been used in previous studies by NASA to evaluate the applications of automation and robotics to the space station; for example, [Bard, 1986].

An evaluation of tactile sensing technologies which depends on criteria ranging from technical elements to commercialization potential does fall under this description. In order to resolve potential difficulties in the interpretation of the initial evaluation, the analytic hierarchy process was applied to the tactile transducer technology for the five highest initially rated technologies, namely 1) capacitive; 2) piezo-resistive; 3) piezoelectric; 4) optical fiber; and 5) optical imaging.

The evaluation criteria were re-organized in a hierarchical framework and local priorities of sub-levels with respect to higher levels were established. One decision was to remove LEO considerations from the evaluation criteria; it was rated an absolute must. Meeting the environmental conditions of the LEO is also flexible in that methods can be devised to protect the technologies from the LEO. At the lowest level, the sensing techniques were assessed against each other for each factor. The comparative process led to eigen-vector solutions to comparative matrices. These solutions were calculated on a computer. The rating factor values used were those of Bard: 1 - equal importance; 3 - weak importance of one over the other; 5 - essential or strong importance ; and 7 - very strong or demonstrated importance.

The results rated the technologies as follows, a higher value indicates a favoured status. Capacitive: 0.2866; piezo-resistive: 0.2682; piezoelectric: 0.1795; fiber optic: 0.1480; optical imaging: 0.1132.

Proximity Sensors

The critical evaluation of proximity sensor technology resulted in capacitive sensor technology and laser triangulation techniques being favored. There were some potential problems with the capacitive technique, one of which being electromagnetic interference (EMI) sensitivity. Another problem is that capacitive proximity sensors are very dependent on the material of objects being approached. In contrast, many of the issues which may reduce the usefulness of laser triangulation techniques have been addressed and corrected for by the DLR (German Aerospace Research Establishment) group [Hirzinger, 1991] and by the National Research Council with respect to its Biris sensor.

The state of the art in miniaturized triangulation laser based proximity sensors are sensors developed by DLR. These sensors achieve precise performance for a wide range, independent of the slant angle and the surface of the measured object.

While the concept for the TPS assumes that the DLR sensors could be used, it is noted that the TPS is modular in nature. In particular, different sensors can be used as long as the data is formatted by the preprocessing electronics in a manner suitable for the rest of the system.

Tactile Sensors

The examination of the TPS requirements favors the incorporation of two types of sensors. Array sensors providing imaging and slip detection capabilities and fingertip force moment sensors for precise measurements of the forces exerted at the fingertips.

Tactile arrays

The capacitive array technique scored the highest during our critical evaluation. It showed several distinct advantages over the other two sensing methods: fewer connecting leads, simpler electronics and the capability to provide a static image of the applied force are only a few examples. However, our studies also put forward the fact that some design difficulties would have to be dealt with, such as EMI shielding and stray capacitance of the sensor, and the need for a compliant dielectric material.

The basic principle behind capacitive sensing is quite simple. It is well known that the capacitance associated to two parallel electrically conductive plates is inversely proportional to the distance between them. When a force is applied on the plates so as to change their separation, a capacitance change results. The latter can be measured and used to infer the associated force.

Ideally, the dielectric material between the plates (compliant layer) should compress linearly in response to the force applied on it. The problem here is to identify a material exhibiting this linear behavior over the proper dynamic range. Silicone rubbers have already shown promising results for that purpose. Although the elastomechanical properties of a plain silicone sheet under uniform pressure are not very good, they can be greatly improved by adding protruding tabs to it.

Capacitive tactile sensors are usually formed into a matrix of several elements. This can be done by attaching parallel conductive traces to the top and bottom surfaces of the compliant layer. By placing the top traces perpendicular to the bottom ones, a capacitor is formed at each intersection. To at least approximate human performance, element spacing under 2 mm is desirable. The materials used in sensor fabrication should allow for mounting on non-planar surfaces such as robot fingers, and should also be compact enough to permit installation in constrained work spaces.

For the TPS, it is proposed to develop a new generation of capacitive based sensors based on prototypes developed in the mid-eighties [Seigel, 1986].

Fingertip force moment sensors

While a taxel array is useful for giving such basic information such as the position of an object, the absolute force on the object is often difficult to determine with any degree of accuracy. This is particularly relevant if the taxel array resolution is very limited. Furthermore, while one can deduce an estimate of the force exerted on an object based on its location and a measurement of torque at the actuator joint, a more accurate and immediate value can be determined if a force sensor is available on the finger. From such a sensor, more stable grasp control loops can be implemented and more complex procedures, such as the determination of an object's hardness through active haptic sensing can be accomplished. A force/moment finger-tip sensor could be implemented (one on each finger) behind each of the taxel arrays.

To measure forces in a practical way, force moment sensors should meet the following minimal requirements: 1) require low actuation force; 2) fit into small volumes; 3) provide high signal-to-noise-ratio (SNR) information; 4) be capable of performing multi-axis measurements; 5) be reliable and modestly priced.

The Field-Effect-Transducer-based approach of SARCOS Research Corporation [S. C. Jacobsen, private communication], incorporated in their six-axis force moment MAST sensor prototypes, promises to be most effective. These are the size of an aspirin and because preprocessing electronics are incorporated in the sensors, only three leads protrude from the unit. Finally since they are fabricated using well known and commercial integrated circuit manufacturing techniques, they should be modestly priced. It is envisaged that these sensors would be ideal for the TPS although strain-gauge based Maltese-cross force moment sensors are also suitable.

Architecture

The tactile and proximity sensor system follows a hierarchical structure as shown in the architecture block diagram (Fig. 1). In brief, the lowest level consists of the various sensors and transducers which gather raw data on the environment and on the state of the manipulator. This level also includes arm and hand actuators. The next level includes the sensor signal preprocessor unit which multiplexes between sensors to acquire desired raw sensor data and conditions it for the next level. On the actuator side, this level incorporates the arm-joint and finger-joint servo-controllers. The next level is the elementary manipulation/identification task (EMIT) control level. It incorporates the EMIT coordination and execution software units which decompose higher level commands from the operator or operating system into lower level device commands with numeric parameters. In order to do so, this unit will access data on the manipulator and environment state from the work-cell model data base and from a sub-level of the EMIT control block, the intelligent signal processor. The intelligent signal processor requests and interprets data from the preprocessor and makes it available to the EMIT coordination/execution sub-level and to the operator interface. Typical information provided by the intelligent signal processor would be the relative angle between the end effector and an object surface. The lower level commands are then sent to the lower sublevel incorporating the manipulator arm and hand controllers. These units transform these commands to actuator commands through kinematic and dynamic transform operations. Information from various levels is available to the operator via the operator interface. The information is also sent to the work-cell model database to update the status of the robot and its environment. Finally, the highest level is a task decomposition and management level which has the capability to decompose a task into a sequence of EMITs.

The EMIT level unit will receive information from the intelligent signal processor and act on the information after comparing this information to expectations based on the work-cell model. Thus if the EMIT receives information that the relative angle between the hand and object is +10° when it should be zero, it will send commands to the manipulator arm and hand controllers to re-orient the hand by -10°. These controllers will act on the servo controllers which in turn will close the loop by acting on the slave arm actuators and hand actuators as appropriate. Robot manipulation will be done under a real time operating system which will govern timing issues and command executions. The EMIT level will receive commands from the operator via human-machine interface software. The EMIT will function in a time scale of milliseconds, well within the capability for an operator to react to new input.

Three operational scenarios are envisaged. The AUTO mode will cause the unit to execute a series of actions automatically although it is planned to allow the operator to initiate the task manually and in fact recover control if he so wishes. The COMMAND mode will allow commands to be executed one by one as decided by the operator. In the MANUAL mode, the manipulator will be under direct control of the operator and only required information, as requested by the operator, will be displayed.

Multisensory Robot Hand

The sensors for the TPS would be mounted on a novel multi-sensor hand as shown in Fig. 2. This hand is an integral part of the SARCOS Dextrous Manipulator (7 degree of freedom (DOF) fully anthropomorphic arm with 3 degree of freedom end effector) which will be available at MPB Technologies' robotic laboratory facilities. The hand will consist of a thumb with two degrees of freedom and two additional fingers providing a parallel jaw movement for closure.

The mid-range proximity sensor for sensing objects at a distance will be installed on the thumb. The 2DOF thumb can be utilized to perform scans and mappings of the work area when requested. This is simpler then moving the entire manipulator and a sweep can be done at relatively high speeds.

At the tip of the other two fingers will be a sensor head containing four close-range proximity sensors, a force moment sensor and a tactile array. These fingers will be used for grasping objects.

This is a unique hand design that will provide information on distances from objects, on shear forces, slip conditions and absolute forces. The proximity sensors offer information ahead and inside the fingers fast and simply. The tactile arrays allow for tactile image processing to assess grasp points and slippage while the force moment sensors on which these will be supported will provide the accuracies on absolute grasp forces and data refresher rates required to achieve closed loop control of forces being applied.

Teleoperation

The evolution of the space station towards a more compact design with significant unmanned periods requires that attention be given to teleoperation from the ground. The sensor-based telerobotic system for a space robot experiment known as ROTEX [Hirzinger, 1991], developed by Dr. Hirzinger's group in Germany, has proven to be a major success. The experiment provided the first demonstration of the use of robotics in teleoperated tasks similar to those envisaged in our strawman tasks. One of the highlights of the experiment was the successful grasping of a free-flying object on the space shuttle via ground teleoperation ***despite a 6.5 sec round trip delay in communications.***

It is possible to have an operator teleoperate the robot from ground in a continuous, manual level. For the operator, the only difference is that information is provided using predictive graphics on the ground and via a stereo TV monitor on the station. The key is to use the world model to predict what sensor information should result from operator commands (either generated by a six axis force/position control ball or by high level command selections) so that a) the operator can visualize the expected robot motion on a high resolution graphics screen and b) send with the commands the predicted values of the sensor data to be generated as these commands are executed. The robot can send back real world updated information and make in situ comparisons between predicted and real sensor data to see if the information is within acceptable parameters or whether the operation should be stopped.

The ROTEX concept allows for the evolution of the system towards more and more autonomy. Through its sensory and teleoperation capability, it also allows for much flexibility in manual operations.

Shared Control

The concepts developed by DLR have been geared to very close *cooperation between man and machine, "shared" autonomy* concepts (gross commands are refined by its local sensors autonomously). Shared control, which is based on close cooperation between man and machines, is in line with findings in industry: fully autonomous robots may not be as effective in enhancing productivity as much as robots which serve to enhance an operator's capabilities but include the operator's decision making capabilities in its control loop.

The shared control concept is based on closing sensory feedback loops locally at the robot's site, thus refining path commands from a human operator handling a force/position control ball or from an intelligent path planner. The operator is able to send crude motion commands and receive time delayed confirmation of command execution. The human operator (via stereo-vision and 3D-graphics) is thus enclosed in the feedback loop on a high supervisory command level but with low bandwidth. High bandwidth sensor-based control loops on board the robot, based on the status of a given task, execute commands and apply fine rapid correction factors based on sensor information of the real local environment.

The shared control concept is exemplified in the connector disassembly task. The gross command "unlock the connector" is supported by local force feedback loops so that the end effector will automatically generate a screwing motion (withdrawal as rotation occurs) to compensate for forces along the rotation axis.

Enhancements for MSS

Unless more tasks can be completed through ground control, staff on station Freedom will be overloaded with regular maintenance activities, leading to what should be considered an unacceptable reduction in mission specialist time for scientific activities. In addition, if ground operation is not implemented, the SPDM is likely to be greatly under-utilized. The TPS will allow more tasks to be completed remotely and lead to faster operations conducted by astronauts. The TPS is also to ensure that manipulation tasks including approach and contact are made smoothly. Although this has not been quantified, this could be important for experiments where the micro-gravity environment must be maintained.

For NASA to allow for an increased workload for the MSS, especially through ground operations will require however several issues to be resolved to their satisfaction:

A viewing and sensing capability for ground operators and robot. The TPS will provide fine control and sensing information (initial contact, touch, and near proximity) to approach objects safely and assess grasp maneuvers. A totally vision-based system may be limited during close-in maneuvers and by communication bandwidth availability.

Latency in control loops and prevention of uncontrolled motion. Ground teleoperation of the MSS will have to circumvent variable data feedback time delays in the order of six to ten seconds. Time delays of this order of magnitude lead to potential control problems and instabilities. With the TPS, fast high bandwidth sensory-based control loops will control dextrous manipulation and positioning. The operator is kept in the loop at a higher level. The sensor-based loops will work to complement operator functions and indicate where discrepancies could cause damage while handling approaching objects. The TPS approach is also based on the end effector position relative to the object to be handled as opposed to absolute positions as stored in a world model. Thus the system also becomes less sensitive to small deviations from what is expected and is less dependent on the exactitude of world models for the space station.

Verification of ground control. A set of instructions can be sent on an up-link and executed. If something unusual happens, the robot is meant to stop and await a new set of instructions. Ground teleoperation control could send not only a set of commands, but also predicted TPS sensor readings resulting from the execution of these commands. These commands could be compared with real TPS readings by MSS computers on board the station to immediately react to any major anomalies and to adapt operations, through local sensory control loops, when only small deviations were detected.

Reliability of robots and fault tolerance of robotics. The addition of sensors which can serve as a back-up to vision and wrist force moment sensors is an advantage in terms of fault tolerance and reliability.

Acknowledgements

This work has been supported by the Strategic Technologies for Automation and Robotics (STEAR) Program, Space Station Program, of the Canadian Space Agency.

References

Bard, J. F., "Evaluating space station applications of automation and robotics," *IEEE Trans. Eng. Management*, vol. EM-33.2, p. 102, 1986.

Hirzinger, G. *et al.*, "A sensor-based telerobotic system for the space robot experiment ROTEX," *Second Int. Symposium on Experimental Robotics*, ISER, Toulouse, France, June 25-27, 1991.

Siegel, D. M., "Contact sensors for dextrous robotic hands," *Master's thesis*, Artificial Intelligence Laboratory, M.I.T., Cambridge, MA 02139, 1986.

Tactile and Proximity Sensor System.

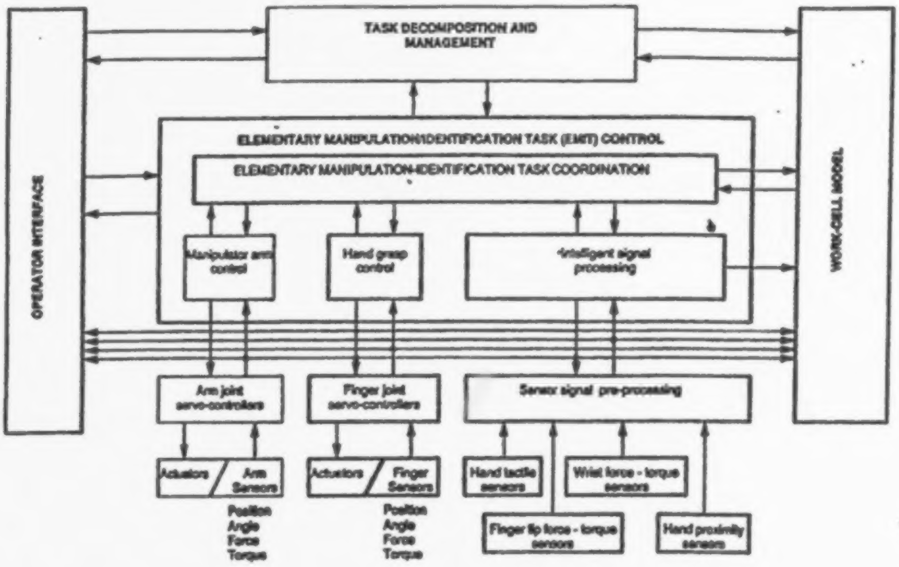

TASK DECOMPOSITION AND MANAGEMENT
OPERATOR INTERFACE
WORK-CELL MODEL
ELEMENTARY MANIPULATION/IDENTIFICATION TASK (EMIT) CONTROL
ELEMENTARY MANIPULATION-IDENTIFICATION TASK COORDINATION
Manipulator arm control
Hand grasp control
Intelligent signal processing
Arm joint servo-controllers
Finger joint servo-controllers
Sensor signal pre-processing
Actuators
Arm Sensors
Actuators
Finger Sensors
Position
Angle
Force
Torque
Position
Angle
Force
Torque
Hand tactile sensors
Wrist force - torque sensors
Finger tip force - torque sensors
Hand proximity sensors

FIGURE 1: Proposed hierarchical architecture for the tactile and proximity sensor system. The EMIT level interprets sensor and work-cell data, as well as commands from the operator and/or task decomposition level to issue instructions to the sensor and robot actuator level.

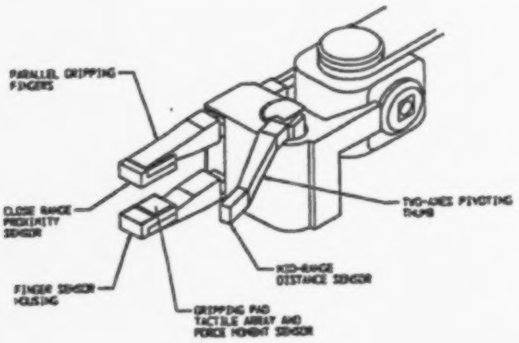

PARALLEL GRIPPING FINGERS
CLOSE RANGE PROXIMITY SENSOR
FINGER SENSOR HOUSING
GRIPPING PAD TACTILE ARRAY AND FORCE MOMENT SENSOR
MID-RANGE DISTANCE SENSOR
TWO-AXES PIVOTING THUMB

FIGURE 2: The multisensory hand consists of a parallel jaw gripper, each gripper finger being equipped with near range proximity sensors, a tactile array and a force moment sensor. A mid-range proximity sensor is mounted on a two degree of freedom thumb.

903228

Track Maintenance Using Real-Time Neural Computing

Jianli Liu and B. Archie Bowen
CompEngServ Ltd.
300-19 Fairmont Avenue, Ottawa, Ontario, K1Y 1X4

#149620

Abstract

This paper discusses an adaptation of the training parameters of a real-time neural net to permit tracking of an input vector (in our case a target) as it moves through the input space. Real-time neural nets such as ART can maintain the identity of recurring input classes of vectors (usually referred to as patterns), can simultaneously identify and learn new input classes, and can forget previously memorized classes if instances fail to re-occur in time. These properties suggest that the ART nets offer a real-time system which, given some adaptations, could be exploited to follow an input vector as it varies dynamically.

The first goal was to maintain the normal ability to identify a new input target, and to modify the behavior of the dynamic training regime to permit the assigned output neurons to stay 'locked' onto moving targets. If this goal was achieved, the second goal was to append some attribute data (slowly changing data, characterizing the input) to the input in order to maintain a 'lock' on the two tracks as they intersect and diverge.

A study was conducted using an ART-1 network to identify and then maintain the tracks of two simulated sonar target signals as their tracks intersected. The experimental results showed that the system maintained both track identity as the targets moved, and, with only a small amount of attribute data, correct tracking identification after the intersection occurred.

Introduction

The acquisition and maintenance of a target from the signal returns (such as SONAR or RADAR) involves establishing the track as it first appears and maintaining its position and identity as it moves, and in some cases re-establishing the track after periods of interference or signal loss.

Real-time neural nets such as the ART nets offer an interesting alternative technique for resolving the track maintenance problem. The ART nets feature as simultaneous and continuous operation and automatic learning (no teacher), and can be made to adapt to changes in the input data characteristics (learning new characteristics and forgetting non recurring inputs). The ART nets are traditionally used in pattern classification applications, and the learning involves identifying recurring new patterns.

It was proposed that given kinematic data, and some innovative changes in the procedures for adjusting the parameters used in traditional classification functions, the ART net can maintain track identity; given, in addition, suitable attribute data, the net can maintain the identity of targets as the tracks merge and separate. An output neuron of an ART is assigned to a new target (simulating a Kalman filter), and it stays locked onto the input vector representing the combined kinematic and attribute vector of the target.

We confirmed the feasibility of this approach through a study funded by the Canadian Department of National Defense. In the study, the sonar target signals were simulated. The resulting system maintained correct tracking identification in each of the experiments. A very small attribute vector was found sufficient to maintain track identity through a variety of trajectories involving target merges followed by a separation.

In this paper, we first present an overview of the track maintenance problem, then introduce the concept of real-time neural computing and the necessary adaptations of the ART architectures, and finally describe the design of the experiments and summarize the experimental results.

Track Maintenance Problem

The track maintenance problem is defined as establishing the identity and maintaining the location of a target from the signal returns. It includes establishing the track as the target first appears as a perturbation in the signal returns, maintaining the target position and identity as it moves through the range, and in some cases re-establishing the track after periods of interference or lack of a signal. The Track Maintenance problem can occur because of the signal-to-noise ratio of the environment, the propagation characteristics of the environment, and in some cases because of direct evasion or confusion tactics on the part of the target.

There are usually two types of information used to characterize a target: kinematic information and attribute information. Kinematic information is a term used to denote data which quantifies the movement or position of a target. Thus angle, range, velocity, etc. are regarded as kinematic data. This information is usually obtained from active or passive sensors. Attribute information denotes data which describes a target's properties, such as reflectivity or the emission of a certain frequency spectrum (in sonar). Normally kinematic data can change continuously with time (such as position), while attribute data is constant or changes slowly over time. Attribute data can be dependent on kinematic data as is the case of when the velocity of the vehicle determines the frequency spectrum emitted by a propeller.

Kinematic data is widely used in classical target-tracking systems. Attribute data has been most often used by experienced operators to form opinions about the target. Skilled operators, for example, can often identify the class of ship by listening to the sonar signals for a period of time. Kinematic data is therefore used to maintain the position and movement of the target while attribute data can be used to determine its identity.

An important part of the tracking problem is to maintain the identification of targets as two or more tracks merge and separate (the kinematic data becomes identical). This is called the track crossing problem. The problem is challenging because kinematic information becomes merged when the two vehicles are within an uncertainty zone in which the two appear as one target and the angles and distances of both targets become coincident. The size of this uncertainty zone depends on the reflectivity of the targets, the sensitivity of the tracking equipment and other factors. When the two vehicle emerge from this uncertainty zone the question becomes - 'which is which?'.

Traditionally, a Kalman filter is assigned when the track is first recognized, and based on continuous predictions is used to re-establish track identity after two targets emerge from the uncertainty zone.

This approach suffers several well known deficiencies. For example, if the tracks do not actually cross, but diverge in some other way from the uncertainty zone, or if the targets stay in close proximity for some time, etc., the Kalman filter may lose its capacity to maintain its tracking ability and a wrong target identity assignment is made.

In some situations it is possible to derive and maintain attribute data from the sensors. Future systems will almost surely provide such information. Given such attribute data, it seems reasonable that after the targets emerge from the uncertainty zone the correct target to track association could be quickly re-established based on this data, before kinematic information is used to follow the track. The goal is to emulate the cognitive processes of experienced operators.

Exploitation of this idea depends on the attribute data staying constant, or nearly so, during the encounter. This assumption suggests that the vector of information describing the target consists of a set of kinematic information, and a set of attribute information. This vector is used to maintain track location and identification. As the targets enter an uncertainty zone the kinematic information merges with the other targets for a period of time and then as the targets emerge, the attribute data can be used to tag the two targets so that track association can be re-established. Clearly the initial assumption is that the attribute data stays constant during tracking and during the transition through the uncertainty zone. A looser assumption is that it varies slowly during the whole tracking exercise.

Real-Time Neural Computing

A real-time neural net is defined as:

> A network of computational elements that performs its function by determining and continuously adapting to the characteristics of input data, without intervention from outside the network.

In most neural net applications in which a non-real-time neural net is used, training and operation of the neural net are separate tasks and completed at two separate times. While this separation has proven successful in developing the neural net applications, it implies, however, several important constraints on the resulting system. The most obvious is the need for a properly defined training set, which contains examples of the data clusters and will lead to the desired operational response. A more serious limitation in some applications is the need to assume that, during operation, the characteristics of the input data set do not change with time.

In real-time neural nets, the changing characteristics of the input data set are learned automatically. More specifically, the nets must learn, without a teacher, the new characteristics of the input data set when they become important and forget the old ones when they become inappropriate. The changes in the characteristics of the input data set over a certain time period will be reflected in the deterioration of the response characteristics of the net. In order to detect such deterioration, real-time neural nets must have the following properties:

1. The operational and training regimes are implemented continuously without a separation of the functions of training and operation.
2. The operational and training equations are functions of time.

The concept of a neuron as both an instar and an outstar [1] provides the basis for real-time neural computing. The concept is based on biological neural systems. The operational and training equations of an instar/outstar neuron are functions of time, each of which represents a series of rising and falling exponentials to account for excitation from the outside world and other neurons, and decay of the output over time. These equations enable the continuous adaptation of a neural network system to stimuli from the outside world.

ART nets, including ART-1, ART-2 and ART-3, are the results of exploiting the instar/outstar concept in a useful architecture [2]. In an ART-1 net only binary inputs are allowed while in an ART-2 gray scale inputs are allowed. Otherwise an ART-2 net functions as does an ART-1 net.

The basic architecture of an ART-1 net is shown in Fig. 1. The net consists of two layers of bi-directionally interconnected neurons. The net also includes a reset and a gain control function.

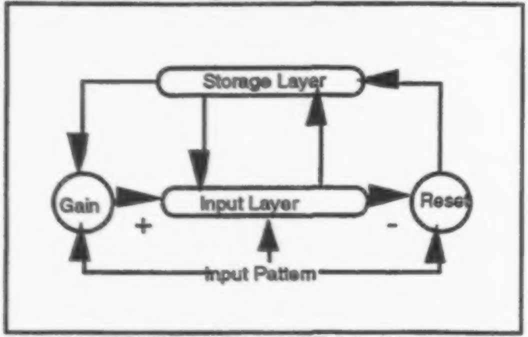


FIGURE 1: Basic ART-1 Architecture

The basic operational and training behavior of an ART net are summarized as follows:

1. A new pattern is presented at the input (this pattern goes to the input layer, the gain control, and the reset system).
2. A global reset signal is activated (since the input layer is inactive) and is transmitted to the storage layer. This shuts off any nodes that may have been active and prepares the storage layer to respond to the input pattern. The gain control is now active (since the storage layer is inactive) and the input layer is therefore enabled.
3. The input layer reflects the input pattern, thereby shutting down the reset system and transmitting the 'bottom-up' pattern to the storage layer (the 'bottom-up' pattern is the input pattern modulated by the weights of the bottom-up inter-layer connections).
4. The nodes in the storage layer compete and finally one of them wins and fires. The winning node transmits back the 'top-down' pattern to the input layer (the 'top-down' pattern is the winning node's output modulated by the weights of the top-down inter-layer connections) and also inhibits the gain control.
5. In the input layer, the match between the 'top-down' pattern and the still-present input pattern is tried. If the mismatch is found, the reset system fires again and the cycle repeats (the winning node is kept off and another node is allowed to win the competition). If the match is exact, the system stops.

Note that, during the operation of the system, learning is going on all the time, so that if iterations are needed to find a match, the weights will have changed over the iteration period. Thus neurons whose weight vectors are far removed from the input pattern at the beginning will move closer as the iterations proceed. Old patterns will tend to be forgotten and the neurons will be assigned to the new pattern.

The ART architectures have several advantages:

1. The system can operate and learn simultaneously and has no need for off-line training.
2. The system can not only learn but also to forget, so that they can adapt to changes in the input data characteristics over time.
3. The significance of input data set does not have to be interpreted by human operators so that the learning is carried out without human intervention.
4. Training does not need to be finished before the system can be used.
5. The order of presentation of the training examples is not important.

The ART architectures also have several limitations:

1. The system has to be carefully designed, in the sense that training control parameters must be assigned, and the input data vector repetition must be synchronized with the internal clock and with changes to the resulting state. There is not a well defined system design methodology available, outside the heuristics developed in particular domain applications.
2. The system is computationally intensive, and in most applications a high performance computer is required. This is, of course, not unusual in any real-time control or signal processing applications.

3. The output layer in both architectures uses so-called grandmother cells (in the output layer) to identify a class of patterns. This single cell is the winner in the competitive learning to characterize a pattern. The result is inefficient compared to a coding system and it is failure prone; obviously, if one cell fails the categorization previously learned is lost.

Despite these limitations, the ART nets' capabilities for real-time operation and learning and the self-adaptation to changes in input data characteristics suggests they are adaptive systems of great potential.

The ART nets are normally used in pattern classification applications. In such applications, the usual behavior of an ART net is as follows: When a new input pattern is received, the net tries to classify it into a known class (one of the previously assigned output neuron). If the input pattern matches the representative pattern of a class (the match satisfies the vigilance constraint of the system), then it is classified into that class. After the classification, the representative pattern of the class is re-evaluated such that the weight vector of the neuron is moved towards the input vector. If the input pattern cannot match any existing class, the net creates a new class, choosing the input pattern as the representative pattern of the class, and assigns an unused neuron to represent it.

When used for track maintenance, it was proposed that an ART net could be considered to treat an input as follows:

> Each track would corresponds to a class containing a single member; When a new target (track) appears, the net will consider it as an unknown and (automatically) assign a neuron to it. When an existing target changes its position, within some limited bounds, the net will adapt the weights to reflect the new position (its normal behavior). The old position of a target will be forgotten and the neuron 'moves along' as the target moves.

The task is to modify the parameters of the ART training regime (vigilance and learning rate) to force the previously assigned neuron to stay locked onto a moving target; and to determine the dynamic behavior of the resulting system.

In an ART-1 net, the similarity of a pattern y to a target pattern x is defined as $S(y|x) = \|y \bullet x\| / \|x\|$, where $y \bullet x$ is the bitwise product of the two strings, called the intersection of the two patterns (similar to the inner product between the two vectors)[2]. The symbol $\|x\|$ is a count of the on-bits in the string. $S(y|x)$ is the quantity which an ART-1 net compares to the vigilance parameter to determine whether to identify the new input pattern with the previously encoded neuron or to assign a new one.

The target position was encoded using a Hamming code. The change between the current position and the next position of a target (the change in kinematic data) will be only one or two Hamming distances. The Hamming distance characterizing the attribute information of the two targets is assumed to be much greater. Thus, the vigilance parameter can be chosen such that when small changes are found in an input pattern, the system will regard them as changes in the kinematic characteristics, so that the identity of the target will not be changed; when large changes are found in an input pattern, the system will regard them as the changes in the attribute characteristics, so that the target will be identified as a new track.

In order to lock an assigned output neuron onto its target, after every classification the old position of the target must be forgotten and replaced with the new position. This implies that a rapid learning (a dramatic change in the 'bottom-up' and 'top-down' connection weights) is required. This can be done by using suitably large learning rates.

Experiments and Results

The goal of the experiments was to provide evidence of the ability of an ART-1 net to fulfill the tracking functions discussed earlier, and in particular, to deal with the track crossing problem.

In designing the experiments we assumed a 32x32 grid of squares designated by the binary equivalent of the x,y position of each square. To maintain a binary distance between adjacent squares, a Hamming code for the input vector was used. Therefore an input vector contained 32 bits for the x position and another 32 bits for the y position of a target, plus certain bits for the attribute data of the target. Targets were assumed to move from square to square, not more than one square at a reception interval.

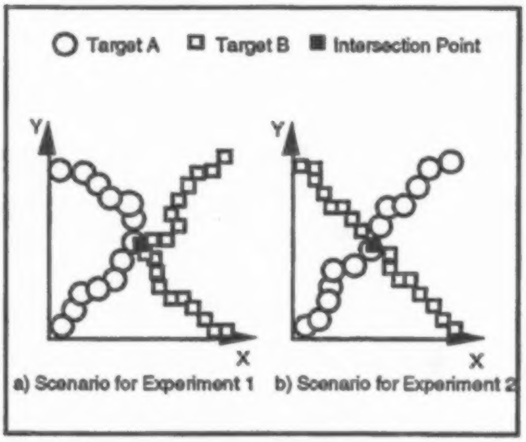
The system was simulated on HNC's neural net simulaiton package (ExploreNet 3000) on a PC-386 with a math co-processor. The input layer consisted of 128 nodes, while the output layer had two. The parameters used to initialize the top-down connection weights were $B1 = 1.2$, and $D1 = 1.0$. The learning rate (the parameter used to set up the bottom-up connection weights) was $L = 1.05$. A vigilance of $V = 0.95$ was selected (this choice is defended later). Two of the experiments illustrate the results:

Experiment 1: Rebound Tracks

Generate two tracks starting at the lower corners and proceeding to an intersection at the center. After intersection, continue the tracks to the top corners above the respective starting points.

Experiment 2: Crossing Tracks

Generate two tracks starting in the lower corners and proceeding to the opposite corners (with an intersection in the middle).



a) Scenario for Experiment 1 b) Scenario for Experiment 2

FIGURE 2: Scenarios for Experiments

The illustrations of the scenarios of the two experiments are shown in Fig. 2. The system maintained correct tracking identification for each of the experiments. Several variations were tried to determine the amount of attribute data needed to maintain track identity. The attribute part of the input vector was arbitrarily reduced from 64 to 32 to 16 and finally to 8 bits; with appropriate changes in the operation and training control parameters, the system continued to track.

The experiments demonstrated a proof-of-concept and confirmed that real-time neural computing paradigms could contribute a new approach to the tracking problem.

Discussion and Conclusion

The work presented in this paper was the results of a study of the feasibility of modifying the usual behavior of real-time neural nets such as ART to resolve the track maintenance problem, in particular, the track crossing problem. The feasibility of the proposed modifications has been confirmed in the conducted experiments.

Many questions still remain at this point. The first question is concerned with the size and the required ratio of kinematic to attribute data. By intuition, we assumed that if the distance of a target's old position and new position was one or two Hamming distances, the minimum attribute vector length of 3 bits should be sufficient to distinguish one target from another. In the experiments, the best results we achieved was 8 bits. This probably implies that we haven't found the best training parameters. The optimal parameter adjustment procedure must be further studied.

The second question concerns the dynamics of the target locking mechanism in a practical application. In the experiments, only two output neurons were employed (corresponding to two targets). This simplified the problem, since an input pattern could only be assigned to one or the other neuron; there was no risk that the input pattern would be mistakenly recognized as a third target. In practice, the system will be placed in a unknown environment in which the number and appearance of targets is not predictable. Therefore, the system must be equipped with more output neurons than required. Then an extra challenge is to avoid a similar target being mistakenly recognized as a new target (assigned to a new neuron), and to avoid an 'over-locking' situation which could prevent new targets from being recognized. This implies that the mechanism of choosing the optimal training parameters becomes much more complex.

One heuristic is based on the following argument.

Let:

R_{same} = The ratio of the minimum Hamming distance between two input vectors representing the same target at two different positions, and the code length of the vector.

R_{diff} = The ratio of the Hamming distance between two input vectors representing two different targets at the same position, and the code length of the input vectors.

D_{same} = The minimum Hamming distance between two input vectors representing the same target at two different positions.

D_{diff} = The Hamming distance between two input vectors representing two different targets at the same position.

Where:

L = The Hamming code length of the input vector
P = The Hamming code length of position part of the input vector
A = The Hamming code length of the attribute part of the input vector
V = Vigilance parameter

From which

$$L = P + A$$
$$R_{same} = D_{same}/L$$
$$R_{diff} = D_{diff}/L$$

Then it was found that:

$$(1 - R_{diff}) < V < (1 - R_{same})$$

This inequality defines the dynamic range in which the vigilance parameter is constrained. It suggests that when the code length of the attribute part of the input vector is shortened, the dynamic range become smaller (the selection becomes more difficult).

If the target can move more than one Hamming distance, then D_{same} could be larger than one; R_{same} becomes larger; $(1-R_{same})$ smaller, and the dynamic range of the vigilance become smaller.

While this is intuitively pleasing, a more detailed study needs to be done.

There could also be some other questions concerned with the issues such as the noise tolerance of both kinematic and attribute data, and other coding mechanisms than the Hamming code.

However, the most important conclusion is that the usual behavior of the ART-1 net can be adapted to track maintenance, if we intentionally and appropriately change the usual procedure of the net's training parameter selection. Considering the real-time characteristic of the ART-1 net, this provides us a very promising alternative technique for the track maintenance problem and other similar problems, which also require real-time adaptation to changes in input data.

The future work could include a further theoretical study of the relation among the vigilance, the learning rate, the kinematic data, and the attribute data; and also experiments using real data to provide conclusive evidence of the applicability of the technique. Another important extension would be to use the ART-2 net to simplify the input data coding problem.

Acknowledgments

The results reported herein were generated as part of a larger project sponsored by the Defense Research Establishment Atlantic (DREA).

References

1. G. A. Carpenter and S. Grossberg, "A Massively Parallel Architecture for a Self-Organizing Neural Pattern Recognition Machine," *Computer Vision, Graphics and Image Processing*, 37, pp. 54-115, 1987.
2. D. S. Newman and T. P. Caudell, "Characterization of Some Learning Properties of ART-1 Machines," *Conference on Neural Networks for Automatic Target Recognition*, Wang Institute of Boston University, May 12, 1990.

90C207

A Neural Network Approach To Detect Traffic Anomalies in a Communication Network

Capt François Viens[1]
Computer Science and Engineering Dept
Collège militaire royal de St-Jean
Richelain, Qué, J0J 1R0.

Dr Nick Dawes[2] **& Dr Bernard Pagurek**[3]
Department of Systems and Computer Engineering
Carleton University, Ottawa, Ontario, K1S 5B6.

#149623

Abstract

LEAPR (Line Element Anomaly Pattern Recognition) is one Knowledge Source (KS) used in a multiparadigm diagnostic system whose goal is to detect anomalies in a Wide Area Network (WAN). The role of the LEAPR KS is to monitor the traffic patterns from the routers and to detect traffic anomalies in a proactive way. The LEAPR KS uses a neural network paradigm called the backpropagation algorithm.

This KS was trained off-line with traffic data from a simulated WAN under normal and abnormal network operations. Abnormal network operations are characterized by different traffic increases and normal network operations are characterized by no traffic increase. After training, this KS was able to rapidly classify in real time the monitored traffic pattern as either NORMAL or ABNORMAL. After diagnosing an ABNORMAL traffic pattern, the LEAPR KS warns the other KSs used in the multiparadigm diagnostic system of an abnormal traffic increase. An abnormal traffic increase might be caused by a congestion or a hardware fault in the network that causes the traffic to be re-routed to a specific line.

This paper presents the LEAPR KS that diagnoses ABNORMAL traffic patterns of any router of the simulated WAN used in the Pegasus research project.

Introduction

As we move into the 90s, communication network management is becoming increasingly complex due to the introduction of new technologies and new services. Some problems generated by this network complexity have been impractical to solve using conventional software techniques. However, Artificial Intelligence (AI) techniques have been proven to be able to solve some of these network management problems [CEBULKA89]. One such AI technique is an Artificial Neural Network (ANN) (also called Neural Network (NNet)).

Artificial neural networks have been proposed as solutions for a wide variety of tasks. Among the most promising is that of pattern classification. Fault detection and diagnosis can be directly viewed as a pattern classification activity, and thus the potential exists for fault detection via ANNs. Trained on shown examples, a neural network will be able to classify new examples. This technique has shown great promise in areas where the domain is poorly understood [CEBULKA89], such as the analysis of traffic patterns in a communication network.

Neural Network

The NNet used for this paper is the backpropagation algorithm which was developed in the 1980s to solve the Perceptron's problem found by Minski and Papert [KNIGHT90]. This NNet paradigm has been found to be useful in addressing problems requiring recognition of complex patterns. It is designed for a multi-layered network operating in a feedforward manner and using the supervised learning mode.

Since the backpropagation algorithm is one of the most commonly used NNet techniques, and because its learning algorithm could be applied to a communication network traffic monitoring problem, the backpropagation algorithm was chosen as the neural network paradigm of choice for this paper.

The training cycle of the backpropagation algorithm is generally very slow. Depending on the

---

1. (514) 358-6622 – Viensf@CMR.CA
2. (613) 788-5730 – Dawes@SCE.Carleton.CA
3. (613) 788-5724 – Pagurek@SCE.Carleton.CA

training data set and on the network architecture, the learning phase can take many iterations. In order to overcome the speed of the learning problem, many variations of the backpropagation algorithm are presented in [KLIMASAUSKAS89]. One of the most widely used variations of the backpropagation algorithm and the one used in this application is the backpropagation algorithm using the MOMENTUM and the LCOEF (Local Coefficient) parameters.

Recent Related NNet Applications

Neural network technology has been applied to many areas [SIMPSON89] such as system diagnostics, handwriting recognition, financial and economic modelling, industrial inspection, and target classification.

Neural network applications in the telecommunications field have been mostly applied as a control or a management tool [KHASNABISH91, CHEN91 and ANSARI91]. A satellite communication network was monitored by a neural network in order to have high network efficiency [ANSARI91]. An adaptive control scheme based on a neural network (backpropagation algorithm) was used to solve a general quality-of-service control problem in broadband Asynchronous Transfer Mode (ATM) networks [CHEN91]. A neural network has been proposed for fast and intelligent processing of transit packets and assigning network access to the local packets in large supra-high-speed packet switched telecommunication networks [KHASNABISH91].

Another interesting neural network application was the use of a neural network as a basic ATM communication network admission controller [HIRAMATSU90]. Based on the traffic and on the service qualities, the neural network decided either to accept or to reject call set-up requests. The interesting feature of this application was that at every control cycle time, based on the history of observed cases, the neural network retrained itself in order to adapt to the new traffic characteristics.

Another paper dealing with network diagnostics is [MAXION90]. This paper written by R.A. Maxion describes an Ethernet anomaly detector based on Local Area Network (LAN) traffic data as the only input. Maxion developed a system that acquires its own internal model of the normal conditions in its environment, and as a consequence, it is able to detect unusual events against a background of normalcy.

Problem

The aim of this section is to present communication network problems that are very difficult to diagnose from the point of view of a network operator. In order to present these problems, the Bell Northern Research (BNR) network is used as a test bed for this paper.

The BNR Network Architecture

The BNR network is composed of Local Area Networks (LANs) interconnected into a Metropolitan Area Network (MAN), and further interconnected into a Wide Area Network (WAN) to other BNR locations around the world [BNR90]. The BNR network spans four continents: North America, Europe, Australasia and Asia.

BNR Diagnostic Problem Description

With over 3000 workstations, 300 bridges and 120 routers, the BNR diagnostic problem [PAGUREK91] can be divided into the three following major problems :

1. Because of the redundancy and multitude of interconnections, there are many possible paths between any two nodes. This causes the network diagnostic operator to generate many diagnostic procedures;
2. With the dynamic routing mechanism, the network can dynamically route packets around apparent faults or congestion. This dynamic rerouting can mask the network problem, and it can also cause overloading elsewhere in the network, leading the network diagnostic operator to an incorrect diagnosis; and,
3. With a high volume of information coming from the Data Collector Device (DCD), it is difficult to get a concise picture of the current situation. Also there can be so much "noise" that the network operators have difficulty determining when the network is starting to behave improperly. In both circumstances, it can be difficult for operators to recognize anomalous patterns and to act before the problem becomes very obvious and costly.

This paper will tackle the second and third of these problems.

The Multiple Paradigm Diagnostic System

In order to detect faults outlined in the previous section, a multiple paradigm diagnostic system seemed appropriate because it reflected the way operators diagnose communication networks using different techniques. For this investigation, a project titled Pegasus, sponsored by the Telecommunications Research Institute of Ontario (TRIO), was initiated. The Pegasus project's aim is to design and develop a multiparadigm diagnostic system that will perform automatic network fault diagnosis [PAGUREK91]. For this multiparadigm diagnostic system, different knowledge sources (KSs) were used and investigated. One of these investigations is the subject of this paper.

The Pegasus Approach

The Pegasus approach is a multiparadigm approach which is composed of many KS integrated through a blackboard system (Figure 1). The main components of the Pegasus project are: the blackboard, the Dragon simulator, and three knowledge sources named: BANES, END and LEAPR.

The Blackboard

A blackboard system was used in the Pegasus project in order to integrate all the knowledge sources (KSs). From the point of view of the blackboard, each KS is seen as a specialist at solving certain aspects of the overall problem [CORKILL91]. One of its main characteristics is that it provides a suitable architecture to develop and test KSs separately. A KS can be added or removed from the blackboard at any time without affecting the other KSs.

The blackboard, which can be visualized as a global database, is subdivided into seven areas, each corresponding to a particular kind of information [IQNEIBI92]. The seven areas are : request, reply, status information, problems, hypothesis, traffic information and finally Knowledge Source Activation Record (KSAR). The last two regions are not currently used and are not illustrated in figure 1.

The blackboard is a strictly passive component and represents a shared memory database. Each KS can independently enquire at one of the blackboard's seven regions to obtain information.

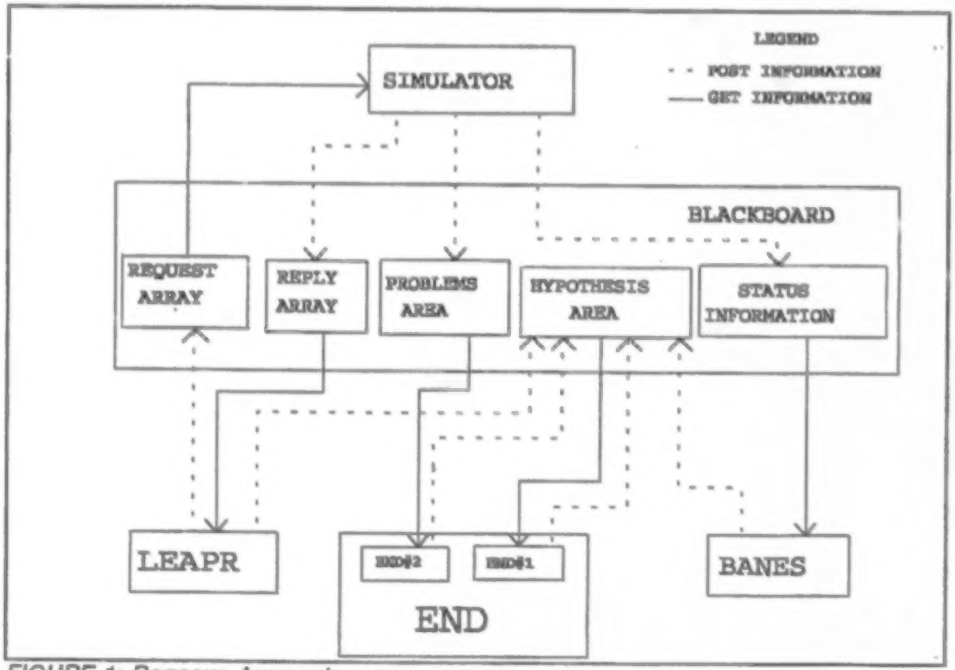


FIGURE 1: Pegasus Approach.

The Simulator

In order to investigate this multiparadigm approach, a simulator that could simulate the BNR network was required. A simulator, called DRAGON, was built by one of the originators of the Pegasus project, Dr. N. Dawes. The DRAGON simulator behaves very much like the real BNR network would under normal conditions by including packet loss, error rates, node faults, data collector reports, and status messages; it even simulates phone calls from users to the network operator in order to report problems [DAWES92]. In the Pegasus project, the DRAGON simulator posts problems and status messages to the blackboard, reads and processes requests coming from any KS via the blackboard, and then posts the replies back to the KS via the blackboard (Figure 1). In order to simulate a long time period, the simulator can run 50 times faster than realtime on medium scale WAN simulations.

The Knowledge Sources

The Pegasus project currently has many knowledge sources to diagnose particular network problems. Out of all the possible KSs, the three KSs outlined in figure 1 are: a probabilistic paradigm approach called BANES, an expert system called END, and a neural network approach called LEAPR.

- The **BANES** (Basic Analysis in Nested Evidence Space) knowledge source takes any set of evidence about some fault and reasons about the evidence using a model of the system. Its reasoning collects beliefs using a new kind of structure called Nested Evidence Space.

This knowledge source monitors the status messages provided by the simulator, and based on its reasoning system, it formulates a hypothesis to the blackboard of any fault it detects (Figure 1). Over extensive trials, this knowledge source detected 98 % of the faults where there was status information correlated to the fault [DAWES92a].

- The **END** (Expert Network Diagnostician) knowledge source is an expert system built to diagnose workstations, routers and routing faults in a communication network [DUSSAULT92]. This knowledge source is written using the COSMIC CLIPS shell, which is a C based expert system tool. END was designed by interviewing the BNR communication network experts to discover their methods of fault diagnosis. Their methods were then translated into rules in COSMIC CLIPS.

- The **LEAPR** (Line Element Anomaly Pattern Recognition) knowledge source uses the neural network approach to diagnose anomalies in the communication network's traffic patterns. This knowledge source is the subject of this paper.

Based on the traffic pattern data on selected lines of a router, the neural network learns what the normal and abnormal traffic patterns are. When these two classifications have been learned by the neural network, this KS classifies the router as either NORMAL or ABNORMAL. When there is an ABNORMAL classification, the LEAPR KS outputs its diagnosis of a router problem to the blackboard. This router problem can then be further investigated by either the BANES or the END knowledge sources (Figure 1).

LEAPR's Justification

As outlined previously, NNet techniques have been used for diagnostic purposes in the medical field, satellite communications and an avionic maintenance system. In the communication network field, NNet technologies have been mostly applied as a control mechanism [KHASNABISH91, CHEN91 and ANSARI91] and have not been fully investigated as a diagnostic tool.

Using a statistical method, an anomaly detector on a LAN with only one traffic input was investigated in [MAXION90]. This anomaly detector uses only one input; and traffic statistics for the different times of the day were used to characterize normal activity. The justification of the LEAPR KS is to generalize Maxion's idea and to use serial traffic data gathered at routers as the inputs to the neural network [PAGUREK91]. The NNet's main advantage over Maxion's work is that a NNet will be able to recognise more complex patterns than Maxion's anomaly detector.

LEAPR Implementation Goal

Due to its pattern recognition capabilities, the NNet is a very appropriate technique to provide early diagnosis of complex NORMAL / ABNORMAL traffic patterns. Early diagnosis of ABNORMAL traffic patterns will allow a network operator to correct situations immediately and to provide very efficient network operations. The LEAPR KS will also act as a proactive agent to signal abnormal traffic patterns to other KSs in the multiparadigm diagnostic system. The other KSs may use this information in their own diagnostic system, and so LEAPR enhances the overall diagnostic system.

LEAPR Prototypes

The goal of this paper is to determine if a general proactive KS could diagnose problems in any router in the simulated communication network. To achieve that goal, four prototypes were investigated before implementing the final LEAPR KS.

Experiment Setup

The prototypes KSs were developed using the following experiment setup:

- Using the DRAGON simulator, a router was selected randomly from among all possible routers for diagnostic purposes. From this router, only lines with a traffic mean above 17 Bytes/second (1,000 Bytes/minute) were considered for diagnosis purposes. Lines traffic mean less than 17 Bytes/second were not considered because their traffic rate was often null, and when their traffic rate was not null, it was varying too much in short periods of time.

- Once the lines were selected for diagnostic purposes, the backpropagation algorithm was trained with examples of NORMAL and ABNORMAL traffic pattern cases. These two traffic pattern cases were generated through the DRAGON simulator. Characteristics of these two cases will be explained later.

- After having trained the NNet using the backpropagation algorithm, the classification knowledge acquired through the trained NNet weights was passed to the KS. At that stage, there was no more learning going on and traffic patterns were ready to be diagnosed by the KS.

After completion of the above three steps, each KS was separately tested with ABNORMAL and NORMAL traffic patterns. The ABNORMAL traffic patterns were simulated by introducing specific traffic increases on selected lines. ABNORMAL cases were characterized by a line traffic deviation of more than 266 Bytes/sec (16 KBytes/min) above its mean. This ABNORMAL case characteristic was assumed to be realistic. The NORMAL traffic patterns were simulated during days of simulated communication network operations where the traffic varies during the day and where no errors were introduced. Figure 2 illustrates the average traffic variation of three lines during one day of simulated network operations: during non-working hours (from 18:00 to 08:00 hrs) the traffic is low, except between 03:00 and 04:00 where there are some management network activities occurring, and during working hours the traffic is high with a decrease in traffic during lunch time. This simulated traffic pattern variation was assumed to be realistic to any typical real network operations.

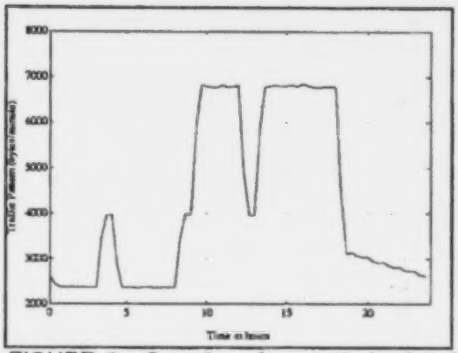


FIGURE 2 : One day of normal simulated network operations.

The Backpropagation Algorithm

In order to capture the knowledge in traffic patterns, a three-layer backpropagation algorithm was implemented in C on a SPARCstation 2. The backpropagation algorithm was verified with independent results from the commercially available NeuralWorks Professional II software package [NEURAL].

The implemented backpropagation algorithm has only one OUTPUT layer Processing Element (PE). Since its output varies between 0.0 and 1.0, for all prototypes in this experiment, NORMAL and ABNORMAL traffic pattern cases were respectively trained with a desired output of 0.1 and 0.9.

Depending on the number of iterations the backpropagation algorithm went through during its training phase, the MOMENTUM and LCOEF parameters, described in the NNet section, were modified dynamically.

In each prototype backpropagation algorithm, the training phase is the most critical part because the backpropagation algorithm learns what is presented to it. If the training set is composed of erroneous data, the backpropagation algorithm will learn these erroneous data. In each prototype, the training set of every backpropagation algorithm had approximately the same number of NORMAL and ABNORMAL traffic patterns. Via trial and error experiments, it was noticed that with a balanced training set (approximately the same number of examples for each category), the backpropagation algorithm results were better.

Also, for each prototype, every backpropagation algorithm was trained with a convergence threshold set at 0.00001. A convergence threshold is a threshold which ends the training phase of a backpropagation algorithm based on the difference between an actual NNet output and a desired output. When the difference (convergence threshold) was less than or equal to 0.00001, the training phase of the backpropagation algorithm was terminated.

Prototype KSs

In order to generalize the KS to be able to diagnose any kind of router, four different prototypes were investigated. The first two prototypes revealed that the KS needed a pre-processing module in front of the neural network (figure 3).

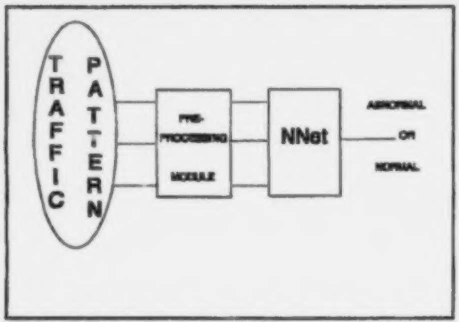


FIGURE 3 : Prototype KS's components.

The pre-processing module calculates the line traffic mean for the last thirty minutes for every monitored line. With this value, the pre-processing module calculates the ratio between the actual line traffic rate and the line traffic mean for the last thirty minutes. Once the line's ratio is calculated, the pre-processing module forwards it to the neural network in order to classify the traffic pattern as NORMAL or ABNORMAL.

LEAPR KS

Based on the prototype KSs, the LEAPR KS was implemented. In order to be able to diagnose any routers in the network, the LEAPR KS is monitoring more than four lines (figure 4).

By comparison to the different prototypes (summarized in Figure 3), the LEAPR KS's architecture is slightly different. The LEAPR KS is composed of a "TOP TWO and BOTTOM TWO" module, a neural network and a Repetitiveness Counter (RC) module (Figure 4).

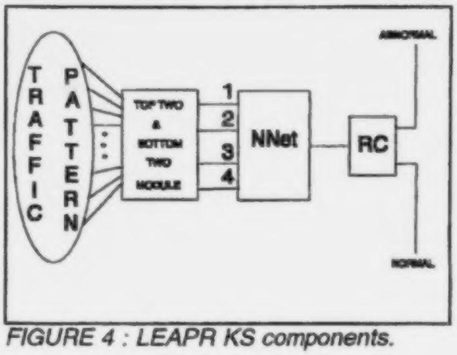


FIGURE 4 : LEAPR KS components.

Before feeding the traffic pattern to the NNet for diagnosis, the traffic pattern goes through the "TOP TWO and BOTTOM TWO" module. This module calculates the line's traffic mean for the last thirty minutes, and calculates the ratio between the line's actual traffic and the line traffic mean for the last thirty minutes. Once each line's ratio is calculated, the traffic ratio is sorted in order to have the biggest traffic ratio as input #1 into the NNet, the second biggest traffic ratio as input #2, the second lowest traffic ratio as input #3, and the lowest traffic ratio as input #4. This method, which considers the two largest and two lowest values, had previously been discovered to be effective in the recognition of rare abnormal events from a mass of normal cases [DAWES92a].

After the "TOP TWO and BOTTOM TWO" module, there is the NNet which classifies the traffic pattern as either an ABNORMAL or a NORMAL case. In both cases, the NNet output is forwarded to the RC module in order to determine the repetitiveness of a categorized ABNORMAL traffic pattern. The way the repetitiveness counter works is that when there is an ABNORMAL case, the repetitiveness counter (RC) is multiplied by a factor of 0.9 and increased by 1.0 (RC = (RC X 0.9) + 1.0). When there is a NORMAL case, the repetitiveness counter is decreased by a factor of 0.1 (RC = RC X 0.9).

The main purpose of the RC module is to reduce the number of false alarms. By monitoring the repetitiveness of ABNORMAL diagnostics from the NNet, the KS is able, based on its RC threshold value, to either accept the NNet's ABNORMAL diagnostic or to reject it. However, beside reducing the number of false alarms, increasing the RC threshold value causes the LEAPR KS to miss some traffic pattern abnormalities; these misses are due to the abnormality either not lasting long enough or not being large enough to be diagnosed as ABNORMAL.

LEAPR Versatility and Performance

The performance results demonstrated that the LEAPR KS could diagnose more than four lines and it could diagnose traffic patterns from unseen routers in the simulated network.

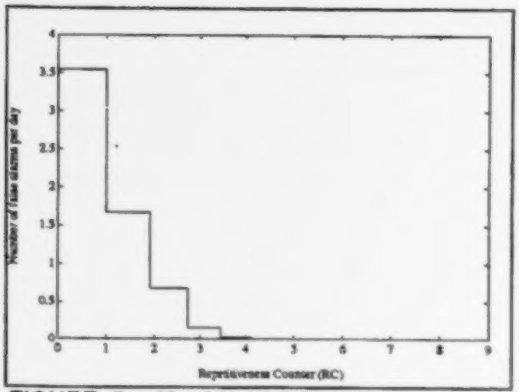


FIGURE 5 : Number of false alarms per day based on a given RC threshold value.

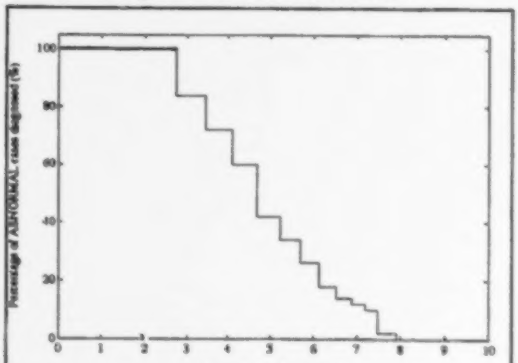


FIGURE 6: Cumulative percentage of ABNORMAL traffic patterns.

In order to test the LEAPR versatility, ten randomly picked routers were monitored during fourteen days of normal simulated network operations, and they were also monitored for a fixed period of time to see if they could diagnose ABNORMAL traffic patterns.

The average efficiency of the LEAPR KS, based on Figure 5 and 6, is excellent. Figure 5 illustrates the average number of false alarms per day and figure 6 illutrates the accuracy[†] of the LEAPR KS. These figures represent the average efficiency based on the results from ten different routers that were monitored by the LEAPR KS. Since the LEAPR is a general proactive diagnostician, for some routers the results were better than others.

Conclusions

This paper, using the backpropagation algorithm as the neural network paradigm, demonstrates that the neural network technology is able to extract the knowledge in traffic patterns in order to classify them in two categories: NORMAL and ABNORMAL. It also demonstrates that the LEAPR KS could diagnose small and complex traffic increases in the simulated communication network. Furthermore, it demonstrates that the NNet technology can be an effective tool in diagnosing traffic patterns in a communication network.

Based on the KS prototypes, the LEAPR KS was implemented as an autonomous KS that could diagnose any router of the simulated communication network (DRAGON).

With the "TOP TWO and BOTTOM TWO" module, the LEAPR KS could accurately diagnose any router of the DRAGON simulator that has at least four lines. This module which pre-processes the traffic data before feeding it to the NNet, makes the LEAPR KS a very general traffic pattern diagnostician. As a general diagnostician, for some routers in the DRAGON simulator, the LEAPR KS detects more false alarms. In order to reduce the number of false alarms, a Repetitiveness Counter (RC) module was implemented, and its threshold value can be adjusted by an operator in order to have fewer false alarms. With the "TOP TWO and BOTTOM TWO" and the RC modules, the LEAPR KS proved to be a very reliable general traffic pattern diagnostician.

[†] **The term accuracy refers to the cumulative percentage of ABNORMAL traffic pattern cases well classified.**

References

[ANSARI91] *"The Performance Evaluation of a New Neural Network Based Traffic Management Scheme for a Satellite Communications Network"*, Nirwan Ansari and Dequan Liu, IEEE Global Telecommunications Conference, GLOBECOM'91, Phoenix, Arizona, December 2-5, 1991, pp 0110 - 0114.

[BNR90] *"bNR Local Area Network Environment - An Introduction for the Network User"*, BNR Information Technology, July 1990.

[CEBULKA89] *"Applications of Artificial Intelligence for Meeting Network Management Challenges in the 1990's"*, Kathleen D. Cebulka, Michael J. Muller, and Christine A. Riley, IEEE Global Telecommunications Conference & Exhibition, 1989, pp 501 - 506.

[CHEN91] *"A Neural Network Approach Towards Adaptive Congestion Control in Broadband ATM Networks"*, Xiaoqiang Chen and Ian M. Leslie, IEEE Global Telecommunications Conference, GLOBECOM'91, Phoenix, Arizona, December 2-5, 1991, pp 0115 - 0119.

[CORKILL91] *"Blackboard Systems"*, Daniel Corkill, AI Expert, September 1991, pp 41 - 47.

[DAWES92] *"DRAGON : A Fast Network Simulator"*, N.W. Dawes, System and Computer Engineering, Carleton University Technical Report SCE 9214.

[DAWES92a] *"A New Statistical Measure of Abnormality and the detection of network faults"*, N.W. Dawes, System and Computer Engineering, Carleton University Technical Report SCE-92-13, 11 June 1992.

[DUSSAULT92] *""A Diagnostic Expert System for Wide Area Network"*, Robert Dussault, Master of Engineering Thesis, System and Computer Engineering, Carleton University, March 1992.

[HIRAMATSU90] *"ATM Communications Network Cont.ol by Neural Networks"*, Atsushi Hiramatsu, IEEE Transaction on Neural Networks, Vol. 1 No. 1., March 1990.

[IQNEIBI92] *"A Blackboard Architecture to support Networks Fault Diagnosis"*, Sami M. Iqneibi , Master of Engineering Thesis, System and Computer Engineering, Carleton University, May 1992.

[KHASNABISH91] "Congestion Avoidance in Large Supra-High-Speed Packet Networks Using Neural Arbiters", Bhumip Khasnabish, Majiid Ahmadi, and Malayappan Shridhar,IEEE Global Telecommunications Conference. GLOBECOM'91, Phoenix, Arizona, December 2-5, 1991, pp 0140 - 0144.

[KLIMASAUSKAS89] *"Volume 1, Neural Computing"*, C. Klimasauska, J. Guiver and G. Pelton, NeuralWare Inc., 1989.

[KNIGHT90] *"Connectionnist Ideas and Algorithms"* , Kevin Knight, Communication of the ACM, November 1990, Vol 33, No 11.

[MAXION90] *"Anomaly Detection for Diagnosis"*, Roy A. Maxion, 20th IEEE International Conference on Fault Tolerant Computing, 1990, pp 20-27.

[PAGUREK91] *"A Multiple Paradigm Diagnostic System for Wide Area Communication Networks"*, B. Pagurek, N.W. Dawes, A.R. Kaye, System and Computer Engineering, Carleton University Technical Report SCE-91-37, 1991.

[SIMPSON89] *"Artificial Neural System, Foundations, Paradigms, Applications, and Implementations"* , Patrick K. Simpson, General Dynamics Electronics Division, San Diego, Pergaman Press, 1990.

[NEURAL] *"NeuralWorks Professional II (Commercial Software)"*, NeuralWare Inc. Penn Center West, Pittsburgh, PA 15276, 1987-1989.

402329

A Neural Network Used to Model the Release of Fission Products from Nuclear Reactor Fuel Under Severe Accident Conditions

William S. Andrews & Brent J. Lewis
Department of Chemistry and Chemical Engineering
Royal Military College of Canada, Kingston, Ontario, K7K 5L0

David S. Cox
Atomic Energy of Canada Limited, Research Company
Chalk River Laboratories, Chalk River, Ontario, K0J 1J0

#149625

Abstract

At present, no comprehensive model exists to predict the release of volatile fission products from Canada Deuterium Uranium (CANDU) fuel in high temperature oxidizing (other than steam), reducing or inert conditions. To address this deficiency, an artificial neural network based on a modified back propagation paradigm has been able to establish a multidimensional correlation to model the cumulative release of fission-product cesium-134 under a variety of temperature, atmospheric, heating and sample conditions. These conditions were used to provide an input space of 14 separate inputs per input vector, with data taken from 29 separate tests of the Hot Cell Experiments 1 and 2 (HCE-1 and HCE-2) and the Metallurgical Cell Experiment 1 (MCE-1). The model was trained with 3662 different input vectors, some of which were reproduced a number of times to produce a balanced training set of 12 516 vectors. The training was conducted over 7 epochs, i.e., complete repetitions of the training set. An independent database of 388 vectors was used to test trained networks. A further data set comprising the results of a complete test not included in the training set was used for network validation.

The networks have been based on the extended delta-bar-delta paradigm contained in the NeuralWorks Professional II/Plus shell. Various hidden layer structures were investigated, with little difference in overall performance found among networks varying from three-neuron single hidden layer structures to ones with fifteen-neuron double hidden layer architecture. Correlation coefficients of .964 for the test set and .995 for the validation set were achieved. Absolute differences between model-predicted and actual measured values, averaged over the complete test set were as low as .05, with overall relative errors of .25. Further, models have been able to reproduce the non-linear response of cumulative release fraction with time. This suggests that a neural network model, which is capable of running in real time, could be embedded into real-time computer codes used for modelling reactor accident phenomena.

Introduction

The accidents at Three Mile Island and Chernobyl have underscored the need to define nuclear power reactor source terms for fission products released under severe accident conditions. It is felt, however, that the accident conditions may be diverse and varied. The atmosphere surrounding the fuel may be steam or air (oxidizing), while the uranium fuel may also experience a hydrogen (reducing) environment. The heating rate of the fuel may vary, but will invariably fall within a wide range, as will the fuel temperature itself. In addition to these considerations, such other factors as extent of fuel burnup, fuel operating history (power level) and location of fuel within the reactor core are also felt to influence the release of volatile fission products from the fuel. The quantitative influence of these factors taken separately or any synergistic effects have not yet been determined with any degree of confidence.

Experiments in the United States involving heating light water reactor (LWR) fuel rod segments have been conducted in steam and hydrogen to attempt to characterize these effects [1]. Analysis of early experiments at Oak Ridge National Laboratory (ORNL) has provided a first-order Arrhenius-type correlation (called CORSOR-M) of the cumulative release of volatile fission products with temperature and time in a steam environment [2]. This model has not been particularly effective in predicting the cumulative fission product release for some subsequent experiments, perhaps in part because it only models two parameters (temperature and time) in one environment (steam). This notwithstanding, the applicability of CORSOR-M to Canada deuterium uranium (CANDU) fuel for pressurized heavy water reactor (PHWR) fuel has yet to be established.

Annealing tests, similar to those of ORNL, have been conducted at the Chalk River Laboratories (CRL) on fuel fragments and mini-elements of CANDU fuel. In fact, the Hot Cell Experiments 1 and 2 (HCE-1 and HCE-2) [3,4] and the Metallurgical Cell Experiment (MCE-1) [5] include a total of 37 separate tests conducted under a wide variety of sample size, conditions and environments. The resulting large database of experimental data has not yet been able to yield an overall comprehensive model to predict experimental results with any degree of confidence.

Some of the molecular-level physical phenomena have been modelled in a mechanistic code, FREEDOM, for CANDU fuel [6]. This model, however, is computation-intensive, and thus does not run in real time. Work continues on simpler, semi-empirical models based on the controlling physical phenomena, such as by Lewis et al. [7] This notwithstanding, the comprehensive model of fission product release considering the full spectrum of conditions of the annealing tests and able to run in real time has not appeared yet anywhere. This paper will outline the development of an artificial neural network (ANN) as an empirical model of 29 CRL tests to predict the cumulative fission product release of cesium-134, a typical volatile fission product.

Artificial Neural Networks

Neural network development began in the mid 1940's, but went into a hiatus due to a lack of discernible applications and sufficiently powerful computers. The most widely-used paradigm, back propagation, emerged in the mid 1980's from the work of two psychologists, David Rummelhart and James McLelland, in their efforts to model the functioning of the brain [8]. Basically, back propagation networks are composed of simple nodes (neurons) which take inputs, sum them, apply a transfer function to this sum and then pass the result on to other nodes. Before the output arrives at the next node, it has a weight applied to it. Current practice is to use three layers of neurons in back propagation networks, as shown in Figure 1. The input layer contains the values of the variables and parameters considered for correlation. For this application, each of the 14 inputs is automatically normalized to the highest value of that particular variable in the input space by the commercial network shell used, NeuralWare Professional II/Plus. Standard back propagation network architecture, as shown in Figure 1, has each input node connected to each node in the adjacent hidden layer. Each hidden node is, in turn, connected to each node in the next layer which, for this application, is a single output node. (During model development, architectures featuring two hidden layers were investigated, as well.) It is also noteworthy that a bias node, with a fixed value of 1, is usually connected to each hidden and output node. The bias node serves to offset the transfer function and tends to cause the network to converge more quickly.

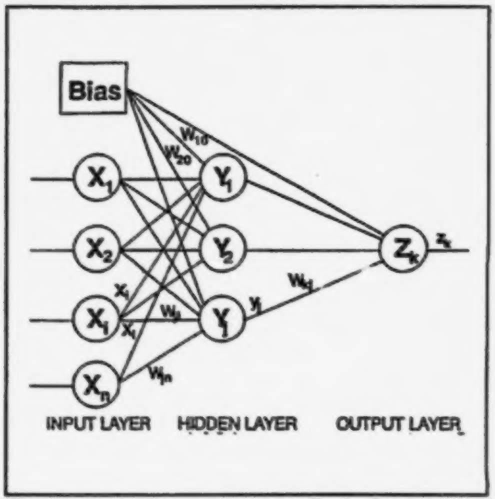


Figure 1: Architecture of a Back Propagation Artificial Neural Network

As shown in Figure 1, the normalized i^{th} input value, x_i, connected to the j^{th} hidden node, has a weight, w_{ji}, applied to it. As a result, the hidden node receives the value $w_{ji}x_i$ as an input from the i^{th} node. With the exception of the input layer, nodes sum the weighted inputs. Thus, for the j^{th} hidden node, the summed inputs can be represented by I_j, as shown in Figure 2, so that

$$I_j = \sum w_{ji} x_i \tag{1}$$

It can be seen that this is also the equation of a linear regression, with the intercept being the weight associated with the bias neuron, w_{j0}.

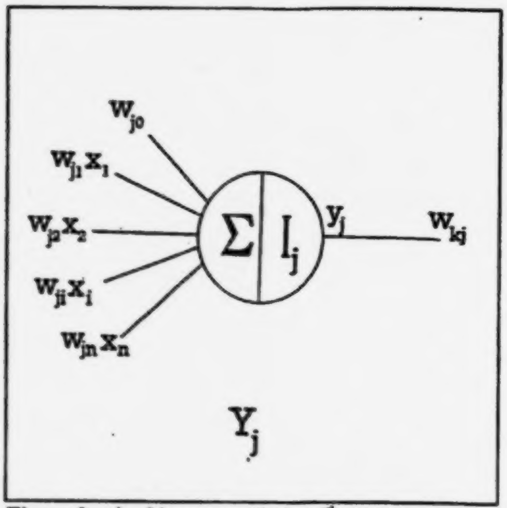


Figure 2: Architecture of the j^{th} neuron in the Hidden Layer.

Non-linearity is introduced into the model by the use of the transfer function, which is applied to the summed inputs, I_j. Although several different functions are available, such as sinusoidal and sigmoidal, the one finding increasingly wide acceptance, and used in this application, is the hyperbolic tangent. The application of the tanh transfer function yields an output y_j from the j^{th} hidden node, such that

$$y_j=\frac{e^{I_j}-e^{-I_j}}{e^{I_j}+e^{-I_j}}. \qquad (2)$$

The advantage of all these transfer functions is that they are continuous and continuously differentiable, with the derivative being expressed in terms of the function itself. This can be seen below for the tanh function.

$$\frac{\partial y_j}{\partial I_j}=(1+y_j)(1-y_j) \qquad (3)$$

In fact, the tanh is the smooth version of the step function from -1 to +1.

This same process is repeated between the hidden layers, if there is more than one, and then between the last hidden layer and the output layer, with the transfer function again applied to the summed inputs to produce the output. When the network is initialized, the values of the weights are randomly assigned. The "knowledge" or "intelligence" within the network, though, resides in the distribution of these weights, which must be adjusted to produce an output as close as possible to the desired output. This process of adjusting weights is called supervised learning and is conducted during the training phase of the network development.

To effect training, the network is not only presented with a full array of inputs, but also with known (measured) outputs for each input set or vector. When the initial pass through the network for a given set of data is complete, an output, z_k, is determined. This value is then compared to the desired output value, d_k, to arrive at a global error E, where

$$E=\frac{1}{2}(d_k-z_k)^2 \qquad (4)$$

The global error is then propagated backwards through the network to adjust the individual connecting weights. This back and forth iterative process is continued until the global error is minimized. At this point, another set of input data (vector) is introduced into the network and the process is continued. Generally, the connecting weights are adjusted after an "epoch", which can be a multiple of the complete training set, where the training set is the set of training vectors, although the epoch does not normally exceed 500 vectors.

Basic back propagation networks use the "delta" learning rule during training, which involves a steepest gradient descent optimization to minimize the local errors, and by extension, the global error. When the global error is propagated backwards, the "local" error at the output, e_k is

$$e_k=-\frac{\partial E}{\partial I_k}=(1+z_k)(1-z_k)(d_k-z_k) \qquad (5)$$

The global error is then propagated back to the hidden layer, where the local error at the output from each hidden node, e_j, is determined by

$$e_j=-\frac{\partial E}{\partial I_j}=(1+y_j)(1-y_j)\sum_k e_k w_{kj} \qquad (6)$$

The adjustment to each connecting weight, Δw_{ji}, (from whence comes the term "delta rule"), needed to minimize the local error, and thus the global error, is determined by the gradient descent rule.

Thus

$$\Delta w_{ji} = -\eta \frac{\partial E}{\partial w_{ji}} \qquad (7)$$

where η is called the learning coefficient, and usually has values between 0.1 and 0.3. This learning rule can be modified by adding a momentum term to take advantage of the preceeding weight adjustment $(\Delta w_{ji})_{old}$, so that

$$(\Delta w_{ji})_{new} = \eta e_j x_i + \alpha (\Delta w_{ji})_{old} \qquad (8)$$

where α is referred to as the momentum coefficient, with $\alpha < 1$ and is often set as $\alpha = (1 - \eta)$.

The commercial shell used to develop the fission product release model provided enhancements to the basic back propagation paradigm which proved very effective. The delta-bar-delta learning algorithm featured a self-adjusting learning mechanism based on the presumption that every connection in the network should have its own learning rate and that learning rates should vary over time. The criteria used for learning rate adjustments are that the rate should be increased if the connection weight changes have the same sign for several successive steps, and decreased if the connection weight alternates sign for several consecutive steps. A further refinement to this algorithm is the extended delta-bar-delta (edbd), which permits each connection weight to also have its own momentum rate, with the momentum rate able to vary with time. The criteria for adjusting the momentum rates parallel those for the learning rates.

During the training phase, input vectors are introduced in a random order and repeatedly until the global error stabilizes and is minimized. At this point, the network is considered to have converged onto the optimal solution.

Once training is complete, the network is tested against data which had not been used during training. Like the training set, the test set should represent, to the greatest extent possible, the whole range of the input space. Also, like the training set, the test set must have known output values available for comparison with the network-predicted values. Finally, the network should be validated by predicting results for a data set representative of a likely application.

Essentially, a trained neural network is an n-dimensional correlation. The link with the physical phenomena it is modelling is through the choice of variables or parameters. Any relationship among these inputs is established by the learning rule itself, and not by any real or postulated physical relationships. This notwithstanding, a careful analysis of the network can permit the various input variables to be ranked in order of their influence on the output.

Development of Input Space

For this application, 14 different inputs were used for network development. Related to Figure 1, then, the input space would extend from x_1 to x_{14}. The initial challenge, however, even before investigating different architectures or network/neuron parameters, was the preparation of the input space itself. Each of the 29 tests used to provide data contained values of temperature and cumulative fractional release measured at intervals of 100 to 300 s. Most tests exhibited a characteristic response, as shown in Figure 3, of an initial plateau due to a low release rate, a steep climb resulting from an increased release rate and, finally, a second plateau due the near depletion of the volatile fission product inventory in the sample. Most of the data available, then, was confined to the initial and final plateaus, with little available for the important high-release-rate portion. Further, more tests were conducted at 1600 °C than at any other temperature, even though isothermal test temperatures ranged from 1350 to 2100 °C. In order, however, for the model to be able to interpolate with any degree of confidence, the input space had to be as balanced as possible. Without this, inadvertent biases would be introduced and trained into the network. In

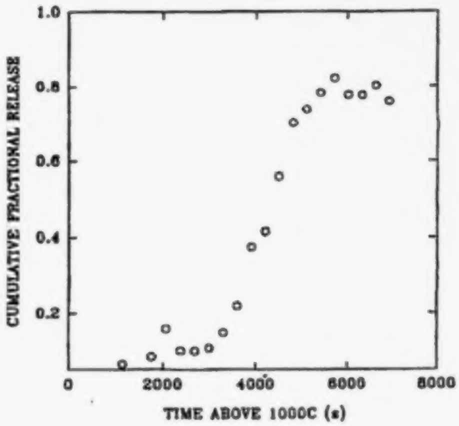


Figure 3: Cumulative fractional release vs. time for CRL test MCE1-1.

other words, the network would tend to provide better predictions for conditions approximating the preponderance of training data and provide poorer predictions for other areas of the input space. To redress this imbalance, the data available for each test were expanded significantly by linear interpolation, so that the available number of training vectors was increased from 1371 to 4049. Any inaccuracies introduced by the linear interpolation were felt to be well within the actual noise of the data itself.

While linear interpolation was used to establish balance in the data within tests themselves, balance among the tests was achieved by repeating some tests. For example, seven tests were conducted at 1500 °C , six of which were in steam and one in air, and two tests were at 2100 °C, both in air. When compared with all the other tests, it was decided, for the seven 1500 °C tests, to repeat the single test in air six times and to use each of the other tests (all in steam) only once, for a total of 1483 training vectors. By comparison, each of the 2100 °C tests was repeated seven times, for a total of 1484 vectors.

Overall, the database of 4049 vectors after interpolation was expanded to 12 516 vectors by repeating a number of tests. An inherent weakness of the database, however, is that complete balance could not be achieved. In the example used above, no data were available for sample performance in steam at 2100 °C. In light of this, the degree of correlation achieved throughout the database is indeed remarkable.

Apart from the training set, a separate test set was required. This was established by taking every tenth value from each expanded data set. This provided a set of 388 vectors which were independent of the training set and yet covered the training set input space. A third data set was created by keeping the uninterpolated data from a complete test (HCE-2 CM6, conducted in steam at 1495 °C) separate from the training and test sets. Thus, the use of the raw data from a complete and "unseen" test would provide validation, or an indication of how well the network could generalize within the range of the input space.

Network Architecture

The initial decision to use a back propagation network was adjusted, after initial success, to the extended delta-bar-delta modification, which converged more rapidly and was more attractive, due to its "self-adjusting" parameters. This still left, however, the question of architecture, e.g., the number of hidden layers and nodes within these layers. In order to satisfy requirements such as ease of analysis of connection weights and simplicity for incorporation into real-time accident progression codes, it was felt the network should be as small as possible. Thus, if a number of different architectures provided comparable results, the simplest model would be the most preferable.

With the number of nodes or neurons per layer, moving from input to output, given as, for example 14-5-1, networks with architectures varying from 14-2-1 to 14-15-15-1 were investigated. Further, the same architecture was used to determine the influence of such other network parameters as the random number seed, epoch size, and number of epochs needed for convergence.

Although the back propagation paradigm was used successfully to model the nine tests of HCE-1, some difficulty was experienced in obtaining convergence over the much larger final training set of 12 516 vectors. Thus, further work on the back propagation was abandoned when the ease of convergence of the edbd paradigm was discovered. Interestingly, edbd networks seemed relatively insensitive to varying network parameters. Different architectures consistently returned comparable correlation coefficients and overall errors. The minimum number of hidden nodes was found to be three, with similar network characteristics found for as many as 15 nodes in each of two hidden layers. Architectures not found successful were 14-2-1, 14-20-1, 14-3-2-1, 14-15-5-1, although further investigation may well have proved fruitful. (Since no advantage would have accrued, it was not pursued.) In fact, none of the factors varied made a significant difference in network performance. Random number seeds, used to control the order of introducing training vectors and initializing weights, were varied repeatedly. Also, the same seed was used on a number of successive runs, and produced slightly different results. Again, epoch sizes were varied from 75 to 215 with no identifiable difference. One run with an epoch size of 12 516, the size of the training set, did not converge. Finally, seven epochs (87 605 vectors) were used for training, although convergence seemed to have been achieved after four to five epochs. The use of seven epochs did not produce symptoms of overtraining, such as divergence of the global error.

Network Performance Assessment

Model effectiveness was gauged in three ways. First, network-predicted cumulative release values (network outputs) for the test and validation sets were plotted against corresponding values actually determined by CRL. If both values were identical, plotted points would fall along a diagonal with a slope of 1 and an intercept of 0. This line provided the datum for comparison with the linear regression through the plotted points.

The second measure was the correlation coefficient, r, (the square root of r^2, which is referred to as the Pearson product moment correlation coefficient or the correlation of determination). The value of r^2 is a measure of the closeness of fit of a scatter graph (in this case predicted vs measured values) to its regression line, with an r^2 value of 1 being a perfect fit. The coefficient of determination can be calculated by dividing the covariance by the sample standard deviations [9]. The r or r^2 value is often expressed in conjunction with the standard error of the mean, which is computed by taking the square root of the sum of the squares of the residuals divided by the root of the sample size.

The final measure was to determine the average absolute and relative size of the residuals. This was felt to be particularly useful, as it provided a direct comparison between the predicted and measured values across the test set.

An example of the scatter plot for a network architecture of 14-5-1 can be seen in Figure 4.

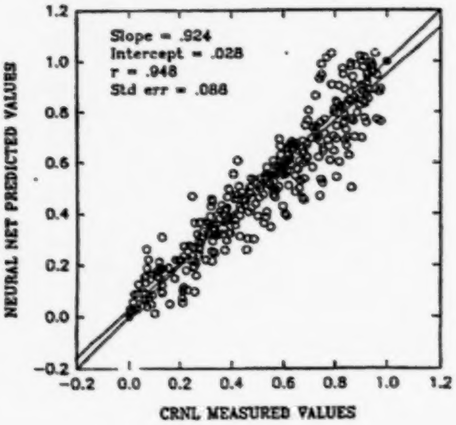


Figure 4: Scatter plot of test set cumulative fractional release values predicted by neural net model vs. values measured by CRL.

The solid dots and the line through them show a perfect correlation, which can be compared to the actual linear regression through the points (the other line). It can be seen that most of the 388 test vectors provide points very close to or on the regression lines. Overall, the slope of .924 and the intercept of .028 are quite close to the optimal values of 1 and 0 repectively. The r value of .948 shows that the regression accounts for 94.8% of the dispersion in the data, with the remainder attributable to the data itself.

The results of the validation of a 14-4-1 network can be seen in Figure 5. In this case, the network provides a very good linear correlation (r is .986). At low fractional releases, the error is as high as 60%, although in this region there is considerable uncertainty in the CRL data itself. In the upper two thirds of the test data, the predicted values are virtually identical to the measured. The standard error was 6.1%, with the average difference between predicted and measured values being .052.

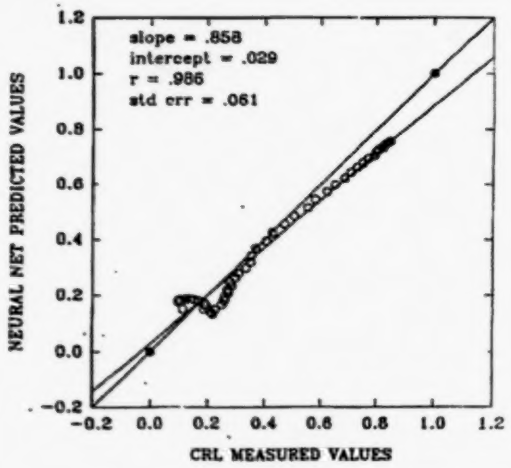


Figure 5: Scatter plot of validation set cumulative fractional release values predicted by neural net model vs. values measured by CRL.

Figures 6, 7 and 8 show the measured and predicted cumulative fractional release plotted against time that the sample is above 1000°C. Of note in all three is that the neural network models provide a smoothing of the data and are able to reproduce the non-linearity of the relationship between fractional release and time, due to varying release rates. Put differently, the networks are able to model the kinetics of the release of volatile fission products under postulated severe accident conditions.

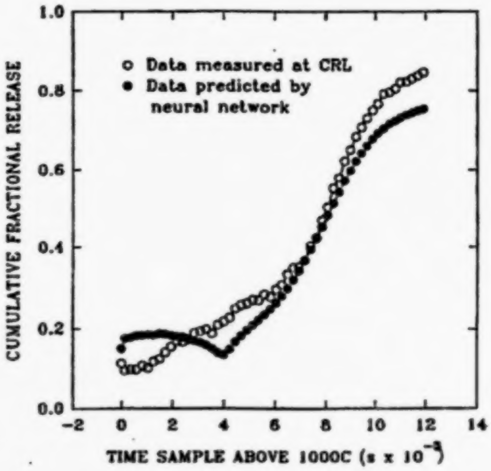


Figure 6: Validation set cumulative release as measured by CRL and predicted by a 14-4-1 neural net model.

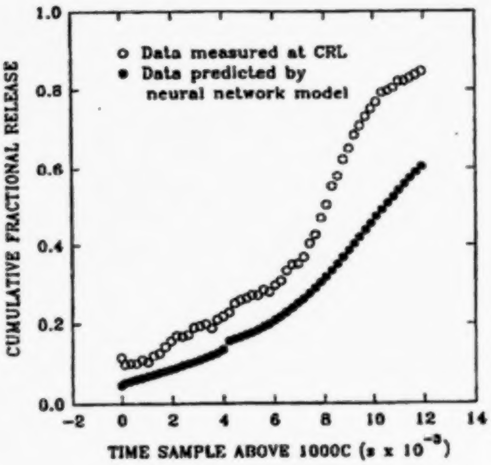


Figure 7: Validation set cumulative release as measured by CRL and predicted by a 14-4-1 neural network with a different random number seed than the network used for Figure 6.

These three figures also illustrate that different network parameters may provide slightly different solutions, due to differing distributions of connecting weights. Figure 6 represents data from the same network as depicted in Figure 5. The discontinuity at 4000 s can be seen in both figures. Figure 7, like Figure 6, represents a 14-4-1 edbd network, but with a different initial random number seed used for training. The discontinuity at 4000 s is still present, but is barely discernible. It should also be noted that, overall, the agreement between predicted and measured data is better in the network depicted in Figure 6 than that of Figure 7, in spite of the rather pronounced anomoly at 4000 s.

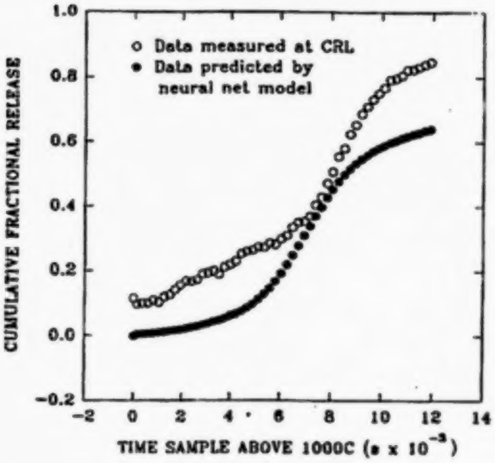


Figure 8: Validation set cumulative release as measured by CRL and as predicted by a 14-5-1 neural net model.

Figure 8 displays the results of a 14-5-1 network with a third random number seed. Here, no anomoly is present at 4000 s, although, overall, the agreement between predicted and measured data is not as good as for the network shown in Figure 6.

Finally, a very useful aspect of the shell used was the ability to conduct a sensitivity analysis of the network. The result is an indication of the relative importance of the various imputs in determining the network output, the cumulative fractional release.

Conclusions

A modification of the back propagation paradigm, the extended delta-bar-delta, with a variety of architectures, has been able to establish a correlation among 14 disparate inputs to predict the cumulative fractional release of the volatile fission product cesium from CANDU fuel fragments and mini-elements under a variety of simulated severe accident conditions. The models were able to reproduce the non-linearities inherent in the relationship between fractional release and time and to provide a smoothing of the data. They were also able to predict the cumulative fission product release fraction of a validation set which had not been used for training within a standard error of 6.1%.

Acknowledgements

The authors gratefully acknowledge the use of the experimental data collected by the Fission Product Release Group of the Fuel Engineering Branch at the Chalk River Laboratories. The experimental work was funded by Atomic Energy of Canada Limited, Ontario Hydro, Hydro Quebec and New Brunswick Electric Power Corporation under the CANDU Owners Group agreement. The work performed at the Royal Military College was funded by the Academic Research Program of the Department of National Defence under allocation 3705-882.

References

1. M.F. Osborne and R.A. Lorenz, "Results of ORNL VI Series Fission Product Release Tests", Proceedings of the 20th Water Reactor Safety Information Meeting, Bethesda, Maryland, 21-23 Oct, 1993.

2. M.R. Kuhlman, D.J. Lehmicke and R.O. Meyer, "CORSOR User's Manual", NUREG/CR-4173, USNRC, Mar 1985.

3. D.S. Cox, Z. Liu, P.H. Elder, C.E.L. Hunt and V.I. Arimescu, "Fission-Product Release Kinetics from CANDU and LWR Fuel During High Temperature Steam Oxidation Experiments", presented at the IAEA Technical Meeting on Fission Gas Release and Fuel Rod Chemistry Related to Extended Burnup, Pembroke, Ontario, 28 Apr to 1 May, 1992.

4. D.S. Cox, Z. Liu, R.S. Dickson and P.H. Elder, "Fission-Product Releases During Post-Irradiation Annealing of High Burnup CANDU Fuel", presented at the Third International Conference on CANDU Fuel, Pembroke, Ontario, 4-8 Oct, 1992.

5. B.J. Lewis, F.C. Iglesias, C.E.L. Hunt and D.S. Cox, "Release Kinetics of Volatile Fission Products Under Severe Accident Conditions", Nucl. Technol., Vol. 99, 330 (1992).

6. L.D. MacDonald, D.B. Duncan, B.J. Lewis and F.C. Iglesias, "FREEDOM: A Transient Fission-Product Release Model for Radioactive and Stable Species", Proceedings of Technical Committee Meeting on Water Reactor Fuel Element Computer Modelling in Steady-State, Transient and Accident Conditions, Preston, England, 18-22 Sep, 1988, AECL-9810, International Atomic Energy Agency, 1988.

7. B.J. Lewis, D.S. Cox and F.C. Iglesias, "A Kinetic Model for Fission-Product Release and Fuel Oxidation Behaviour for Zircaloy-Clad Fuel Elements Under Reactor Accident Conditions", presented at the International Nuclear Congress 93, Toronto, Ontario, 3-6 Oct, 1993.

8. D.E. Rummelhart and J.L. McLelland, "Parallel Distributed Processing: Explorations in the Microstructure of Cognition Volume 1: Foundations", MIT Press, Cambridge, 1986.

9. R.L. Mason, R.F. Gunst and J.L. Hess, "Statistical Design and Analysis of Experiments with Applications to Engineering and Science", John Wiley & Sons, New York, 1989.

906228

Autonomous Impedance Control of a Robot in an Unstructured Environment Using a Qualitative Task Model

S.P. Ananthanarayanan, Z. Lu, A.A. Goldenberg
Robotics & Automation Laboratory
University of Toronto, Toronto, Ont.,
M5S 1 A4, Canada.

J. Mylopoulos
Department of Computer Science
University of Toronto, Toronto, Ont.,
M5S 1A4, Canada

#149627

Abstract

This paper discusses a new approach to autonomous control of constrained manipulation using qualitative modelling and reasoning techniques. By constructing qualitative representations of the task, environment and the basic laws that govern the physical world in a conventional low-level controller, the robot is provided with a capability to counteract large uncertainties in the environment by employing qualitative (common-sense) reasoning. An impedance controlled constrained manipulation task is considered as an example, and an experimental demonstration of how this low-level controller is enhanced in its capabilities to cope with large task uncertainties by introducing a complimentary qualitative task model and reasoner, is illustrated in a step-wise manner.

1. Introduction

Traditional low-level robot controllers that have been designed using analytical formulations of reasoning and analysis [1,2,3] have serious problems in dealing with uncertainties and/or anomalies that plague the physical world. The inherent specificity in detail required by these analytical formulations (which are desirable for achieving accurate control), prove to be detrimental in the wake of significant disparities between the 'assumed' (ideal) model and the actual world. In particular, these problems are more pronounced when the task to be performed by the robot is highly complex. In order to enable the low-level robot controller to counteract with uncertainties and still achieve successful performance of the task, a (coarse grain, qualitative reasoning based) supervisory controller has been proposed as a supplement to the existing (fine-grain quantitative reasoning based) low-level controller. This supervisor can 'guide' the low-level robot controller through large unmodelled uncertainties or unforeseen circumstances, by appropriately changing one or more of the control parameters and/or the command trajectory on-line, based on a 'common-sense' assessment of the situation. The qualitative supervisor mentioned above, uses a qualitative model of the task, robot and the environment to perform 'human-like' first principles reasoning [4,5,6].

In general, a *qualitative model* of a given manipulation task, symbolically abstracts the mathematical model of that task. The principle in the development of a qualitative model is to classify the range of values that a state variable in a mathematical model can assume into a finite, totally ordered set of physically meaningful qualitative states. Since qualitative synthesis uses these finite set of qualitative states, complexity in the synthesis of the task is greatly reduced. More explicitly, the qualitative model will have all the elements comprising a multi-dimensional Eucledian vector space represented qualitatively [6]. This includes qualitative representations of Cartesian Coordinate Frames, vector quantities that are relevant to robotic modelling such as force, displacement, etc. For instance, a qualitative force is represented by two parameters; (i) a qualitative magnitude that maps an interval in the quantity space of the magnitude of the force, and (ii) a three dimensional qualitative direction vector that specifies the direction of action of the force by means of three symbols, -,+ and 0. These forces have also been represented as qualitative wrenches in "qualitative screw coordinates" and the motions as qualitative twists. Further a symbolic algebra has been defined that includes scalar addition and multiplication, and vector addition and product, very similar to conventional algebra. Properties such as, reciprocity have been defined between qualitative wrenches and twists to ascertain qualitatively the constraint conditions imposed by multiple forces acting on a body. In addition a qualitative representation of first principles knowledge of center of mass, gravity, equilibrium, constraint, etc. are defined by appropriate predicates. Using these formalisms, the qualitative reasoner can generate the behaviour of the system (called qualitative simulation [4,6]), similar to QSIM [4], for any given input (qualitative) state vector and also predict the behaviours in future states. Also it can suggest remedial actions to the low-level controller on line, in case the future behaviour of the system is predicted to be undesirable.

A general discussion and development of strategies for qualitative synthesis, analysis and simulation of complex manipulation tasks has been provided elsewhere [6]. This paper deals with a specific application of this approach - the development of a qualitative supervisor that can provide on-line error recovery and guidance to a conventional mathematical model based robot controller, in the presence of large task uncertainties. The qualitative and the quantitative formalisms shown in fig. 1 constitute an intelligent controller. More specifically, an impedance controlled surface painting task has been considered in this paper as an example.

The paper is organized as follows. In section 2, the implementation of qualitative supervisory control for a surface painting task is explained. Section 3 concludes the paper.

2. Implementation of Qualitative Supervisory Control

Let us assume that a stable low level impedance controller[1] is available for performing the task of surface painting. For qualitative supervisory control of this task, it is necessary to embody the knowledge of the underlying functional relationship between the impedance of the robot and the admittance of the environment. That is, the supervisor must 'know' that along a constrained direction of a 2 dof robot shown in fig. 2 (X-axis), $I_r * \delta x_r = F_{xe}$, where, I_r is the impedance[2] of the robot, δx_r is the distance between the commanded and the actual position of the robot (see fig. 2) along the constrained direction, and F_{xe} is the force exerted by the environment as measured by the force sensor. In the case of a rigid wall, F_{xe} proportionally increases with an increase in δx_r and therefore, in order to reduce F_{xe} either I_r or δx_r or both must be reduced, depending upon the case (linear assumption). Similarly, an increase in I_r or δx_r consequently increases F_{xe}. Likewise, along the unconstrained (free) direction $A_r * F_{ye} = \delta y_r$, where A_r is the admittance of the robot along the unconstrained direction, F_{ye} is any external force acting on the robot and δy_r is the motion trajectory error. Since F_{ye} must be rejected by the system leading to a minimal δy_r, A_r should be as small as possible. A physical 'contact' of the robot with the environment is asserted by the qualitative supervisor along x or y directions if the magnitudes of the forces F_{xe} or F_{ye} respectively, are above a certain threshold value F_{min} for a sufficient period of time $\delta t > \varepsilon$ where ε is a known integer. A loss of contact along a particular direction x or y is interpreted by the supervisor if the magnitudes of F_{xe} or F_{ye} respectively, are below F_{min}. A complete loss of contact is envisioned if both F_{xe} and F_{ye} are below F_{min}. The qualitative supervisor uses the *principle of minimum energy* in order to generate its on line refinement actions in the wake of anomalies. This is explained below. We can establish a potential field and define an equilibrium configuration[3] for the robot as one that has a stationary value for the potential. The function that expresses the instantaneous potential for the robot under consideration is given by,

$$I_x\,\delta x_r^2 + I_y\,\delta y_r^2 = 2P \qquad (1)$$

where I_x is the impedance of the robot along the X-axis, I_y is the impedance of the robot along the Y-axis, δx_r and δy_r are the respective distances between the command and the actual position of the robot as stated earlier, and P is the potential energy of the robot at that point. Note that set of all $\{\delta x_r, \delta y_r\}$ that give rise to the same P represent an ellipse of equipotential points in the field (see fig 3). Therefore, the instantaneous coordinates of the command position is represented by the coordinates of the center of the ellipse and the instantaneous current position of the robot is represented by a point on the periphery of the ellipse.

The center of the ellipse represents a point of zero potential. A point inside the ellipse in fig.(3) is at a lower energy level than one on the periphery and therefore, the impedance controlled robot will always have a tendency to move into the ellipse (from the periphery) towards the center (command position) which has zero energy. Thus if the instantaneous task objective is located at the center of the ellipse, the instantaneous motion of the robot will always be towards the task objective. By either regulating the respective impedances (altering its major and/or minor axis) or regulating the distances δx_r and δy_r (changing the size of the ellipse) the qualitative supervisor changes the behaviour of the robot. However, these parameters can not be changed arbitrarily. They can be regulated only within a range of physical values that are possible to achieve which are made known to the qualitative supervisor.

---

[1] It is not necessary that only an impedance controller has to be used. Any low level control methodology can be used. However, the design of the qualitative supervisor will be different for each case.

[2] Here it is assumed without any loss in generality that the impedance is a stiffness and the admittance is a compliance.

[3] By this statement we tacitly assume that the non-conservative forces are neglegible. If the non-conservative forces like for instance kinetic friction, are not negligible, the system will approach the the equilibrium position of minimum energy but will never quite reach it.

An experiment was performed to demonstrate such a capability in a qualitative supervisor designed to monitor a five-bar direct drive mechanism (robot) following a trajectory AB on a wall (see fig. 4) mounted in its workspace (simulating a surface painting task). The impedance parameters of the robot were chosen such that the robot had a small impedance along the X-axis to generate the desired forces needed for the task and a large impedance along the Y-axis to comply with the requirement of accurate trajectory following along the Y-direction. The above rationale was represented in the form of rules. For instance, a typical example of a rule is the following:"if positional accuracy along *Y-direction* is *high*, *Y-Impedance* is *stiff*. Note that the *Y-direction* here referes to a qualitative direction that is defined with respect to a qualitative reference frame, which is an abstraction of a Cartesian Frame of reference. Also, the terms *high* and *stiff* correspond to specific landmarks in the quantity spaces of *positional accuracy* and *Y-Impedance* respectively.

A large bump was introduced along the trajectory of the robot (as a manifestation of the uncertainty). The location and size of the bump were unknown to the robot (fig. 4). As a first experiment, the robot was asked to follow the trajectory AB (which includes the unknown bump) using a stand alone low level impedance controller. On encountering the bump the dynamics of interaction between the robot and the environment changed. The robot was no longer unconstrained along the Y-direction. The external force F_{ye} along this direction began to rise sharply due to the increase in δy_r caused by the instantaneous constraint to the motion along this direction. The low level impedance controller generated very large torques and aborted its operation.

As a second experiment we introduced a qualitative supervisor as a complement to the low-level controller, such that together, they represented an intelligent control architecture (as shown in fig. 1). The objective of this experiment was to demonstrate how such a qualitative supervisor can counteract with this task uncertainty by autonomously reasoning from first principles and trying out alternate strategies in a systematic step-wise manner.

As in the previous case, on encountering the large bump (shown in fig. 5 in top view), the robot got stuck at point R and there was a substantial increase in the external force. However, when F_e crossed a certain threshold value F_{min}, the supervisor 'identified' F_{ye} as a force exerted by a constraint along the Y-axis.The supervisor attempted to increase the potential at R such that there is a point in the periphery of the bump that is located inside the instantaneous ellipse at R. There were three options for the supervisor to realize the above viz., (i) to change the admittances along the X and Y-axis (ii) to alter δx_r (iii) to alter δy_r. Each option had its own implications. Also, a certain order had to be followed in case the supervisor had to exercise more than one option. The supervisor performed qualitative reasoning of the situation as follows.

The first option of the supervisor was to allow the center of the ellipse to move away from o to o' along the trajectory shown in fig. 5. This would cause an increase in δy_r and thereby an increase in the robot's capacity to move from R, as the area of the ellipse at R becames larger. However, this increase could only be done within certain limits in order to not violate physical constraints such as saturation, since an increase in δy_r also implied an increase in F_{ye}. In figure 5, the farthest the center could be moved without reaching saturation is shown as o'. The supervisor terminated the increase of δy_r at this point, since it had a prior knowledge of this limit. Note that, if the surface of the bump at R had been included in the area RP'Q (in fig 5), the robot would have instantaneously moved from point R along the tangent to R. However, from the fact that the robot did not move from the point R, the qualitative supervisor deduced that the bump is much 'steeper' at R than that can be handled by a mere increase in δy_r.

As a second option, the supervisor attempted to decrease δx_r[4]. This was done by changing the command trajectory along the X-axis. This is depicted in fig.6. As the command position along the X-axis is moved incrementally along the negative X-direction below the x-coordinate of point R, the robot started moving instantaneously along the tangent to R as shown by the arrow in fig. 6. Note that the local surface of the bump at R is inside the ellipse with center o''. If the center is moved further down, the robot will have a tendency to move along the surface of the bump. However in this pursuit, if the center of the ellipse happens to fall outside the bump, the robot will lose contact, which is undesirable. This imposes a restriction on how far the supervisor can move the command position along the X-axis. Since the bump is large, the supervisor encountered this problem. Therefore, the supervisor stopped moving the command trajectory when the the contact forces along both directions became lower than a threshold F_{min}, which forewarned a complete loss of contact. In general, this can happen either (a) before the robot reaches the peak or (b) when the robot is at the peak or (c) when the robot has crossed the peak. In any case, the supervisor was forced to reverse its action viz., to increase the force along the X-axis[5] by moving the command position along the positive X-axis, into the constraint . This action can result in any of the three possible instantaneous motions of the robot depending upon the position of the robot on the bump viz.,

(i) if the increase in δx_r is done prior to reaching the peak, the robot will retrace its path back to R (see fig. 7)

(ii) if the increase in δx_r is done exactly when the robot is at the peak, the robot will get stuck at the peak (see fig. 8), and

(iii) if the increase in δx_r is done after crossing the peak, the robot will crosses over the peak without losing contact

Since the geometry of the bump was not known, the supervisor had to infer whether it had crossed the peak only from the above response. The ideal response is of course the last of the above-mentioned (item (iii)). In this case, the problem is solved. However, if the reaction of the robot is either of that mentioned in item (i) or item (ii), the supervisor concluded that the increase in δy_r (made initially) was not sufficient to push the command trajectory beyond the center line (right hand side of the peak as in fig. 9) of the bump along the Y-axis. This is essential because, unless the command trajectory is on the right hand side of the bump the robot will not be able to cross over. It is worthwhile to recapitulate here that, the current δy_r already corresponds to the maximum allowable force F_{ye} ($F_{ye} = F_{max}$) . Therefore, the supervisor can not counteract with either of the reactions of the robot enumerated in items (i) and (ii), with a further direct increase in δy_r. Prior to increasing δy_r, the admittance along the Y-axis has to be changed. Note that an increase in admittance along the Y-axis will facilitate a larger δy_r for the same maximum limit of F_{ye} ($= F_{max}$). In other words, a change in the admittance changes the shape of the ellipse (fig. 10). However, the magnitude of admittance regulation is unknown to the supervisor, once again, because of the lack of knowledge of the geometry of the bump.

Thus, if the robot retraced to the point R, the supervisor increased the admittance along the Y-axis by a small increment and repeated the same procedure from the beginning viz.,

(i) incrementally increasing δy_r (until the forces along Y-axis reach a maximum limit)

(ii) if the robot does not respond to step (i), moving the command position away from the constraint along the negative X-axis (fig. 6) until the forces along both X and Y fall below a minimum, forewarning a loss of contact, and

(iii) if loss of contact is envisioned in step (ii), moving the command position into the constraint in order to increase the force along the X-axis .

If the robot retraced its path back to R once again, a further incremental increase in the admittance was made and the same procedure was repeated. Thus, by a systematic trial procedure, the supervisor arrived at a value of the appropriate magnitudes of the admittance, δxr and δyr that are *just* sufficient to lead the robot over the bump without violating contact constraints.

---

[4] The reason for not choosing this option at the beginning is because such an attempt may give rise to the possibility of losing contact. Since the geometry of the bump is not known, a sufficient amount of force along Y-axis has to be built up (to be on the safe side) before an incremental change in δx_r can be applied.

[5] The supervisor can not increase the force along the Y-axis because it is not sure if such an action would also cause the center of the ellipse to fall outside the bump . This concern is especially valid near the peak of the bump.

Two different sizes of bumps were introduced (see fig 12). The robot crossed over the first one (the larger one) after three retracings and corresponding variations in the admittance. In the case of the small bump, the robot crossed over without losing contact in the first attempt itself, by just applying the trajectory regulation procedures, without having to regulate the admittance. Figures 13, 14 & 15 show the performance of this task. From fig. 13 (a) and (b), a comparison can be made between the two cases shown : with and without a qualitative supervisory controller. It is clearly seen that the forces along the Y-axis in fig. 13(a) increase astronomically when the supervisor is not present. When the low level controller is complemented with a supervisor, it can be seen that the force along the Y-axis is not allowed to exceed a limit (of 30 N in this example). The three trials for getting over the large bump are shown explicitly in the figures 13 and 14 respectively. It is also evident from fig. 14 (b) that the robot does not lose contact at all, as the resultant contact force is never zero or less in the whole operation.

3. Conclusions

There are a variety of manipulative tasks such as grinding, deburring, surface painting, glass-cleaning etc., wherein both forces and motions will have to be controlled. In these tasks, if a large uncertainty exists in the environment, the controllers can easily degrade in performance. An experimental demonstration was provided in this paper on how large uncertainties in the environment can be handled successfully by a qualitative reasoning supervisor. This qualitative reasoner, which uses first principles knowledge of the mechanics of the task and the physical laws of the world to 'monitor' and 'guide' the low-level controller under anamolous situations, attempts to imitate human behaviour under similar circumstances. Further, this approach paves a way to autonomously regulate the impedance parameters in a task to a desired value, when these parameters are only approximately known or chosen.

References

[1] G. J. Liu, A. A. Goldenberg, Experiments on Robust Control of Robot Manipulators, Proc. of 1992 IEEE International Conference on Robotics & Automation, France, 1992, pp. 1935-1939

[2] R. Volpe, P. Khosla, An Experimental Evaluation & Comparison of Explicit Force Control Strategies for Robot Manipulators, Proc. of 1992 IEEE International Conf. on Robotics & Automation, pp. 1387-1393

[3] Z. Lu, A Sliding Mode Based Impedance Control and Force Regulation, Ph. D dissertation, University of Toronto, Canada, November 1992.

[4] B. Kuipers, "Qualitative Simulation", AI-29, pp- 289,1986.

[5] Forbus K.D., "Qualitative Process Theory", AI-24, pp- 85,1984.

[6] S. P. Ananthanarayanan, A Qualitative Framework for Intelligent Control of Dexterous Manipulation, Ph.D. dissertation, University of Toronto, 1993.

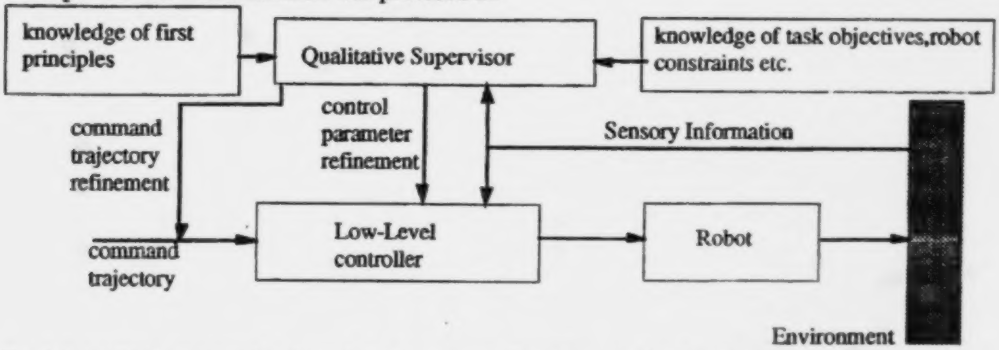

knowledge of first principles
Qualitative Supervisor
knowledge of task objectives,robot constraints etc.
command trajectory refinement
control parameter refinement
Sensory Information
command trajectory
Low-Level controller
Robot
Environment

Fig. 1 A qualitative supervisor in an intelligent controller for constrained manipulation tasks

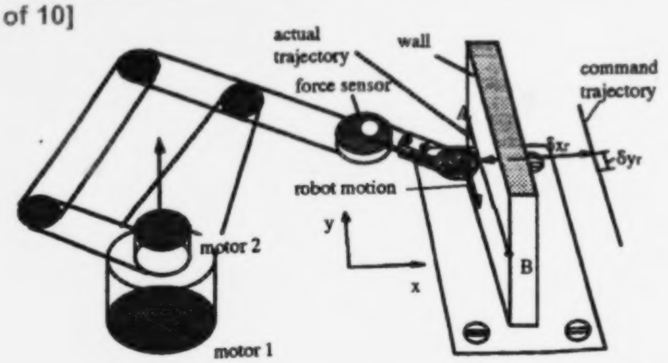

actual
trajectory
wall
command
trajectory
force sensor
A
δxr
δyr
robot motion
y
x
B
motor 2
motor 1

Fig. 2 The interaction between the environment and the robot

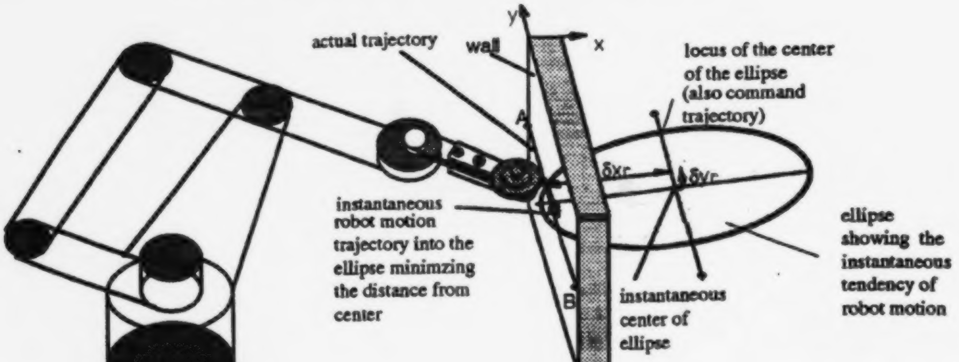

y
x
actual trajectory
wall
locus of the center
of the ellipse
(also command
trajectory)
A
δxr
δyr
instantaneous
robot motion
trajectory into the
ellipse minimzing
the distance from
center
ellipse
showing the
instantaneous
tendency of
robot motion
B
instantaneous
center of
ellipse

Fig.3 The principle of minimum energy

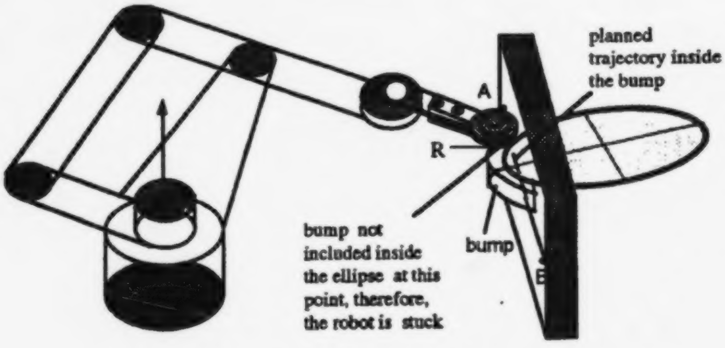

planned
trajectory inside
the bump
A
R
bump not
included inside
the ellipse at this
point, therefore,
the robot is stuck
bump
B

Fig.4 An illustration of an uncertainty in the task

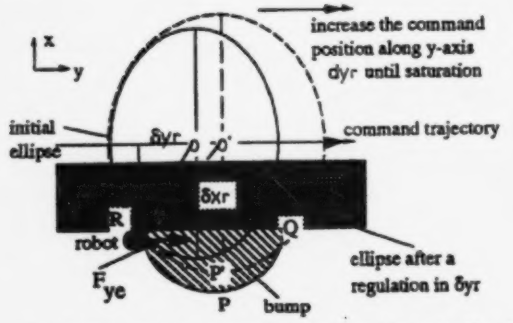

x
y
increase the command
position along y-axis
dyr until saturation
initial
ellipse
δyr
o
o'
command trajectory
δxr
R
robot
Q
Fye
P'
P
bump
ellipse after a
regulation in δyr

Fig. 5 Changing the command position along the Y-axis

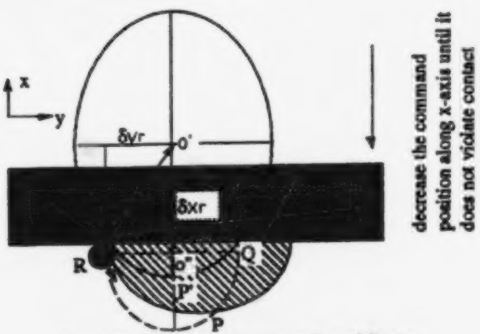

x
y
δyr
o'
δxr
R
Q
o''
P'
P
decrease the command
position along x-axis until it
does not violate contact
constraints

Fig. 6 Reducing the command position to counteract with the `steepness' of the bump

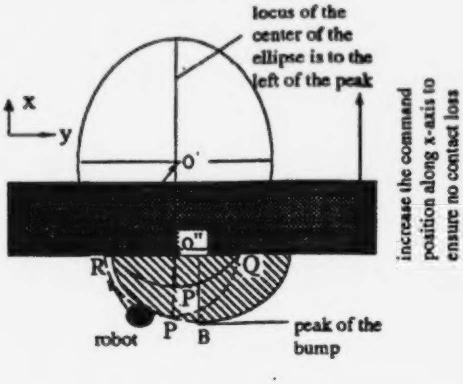

locus of the center of the ellipse is to the left of the peak
x
y
O'
O"
R
Q
P'
P
B
robot
peak of the bump
increase the command position along x-axis to ensure no contact loss

Fig. 7 An increase along X-axis causes the robot to retrace -the ellipse center is to the left of the peak

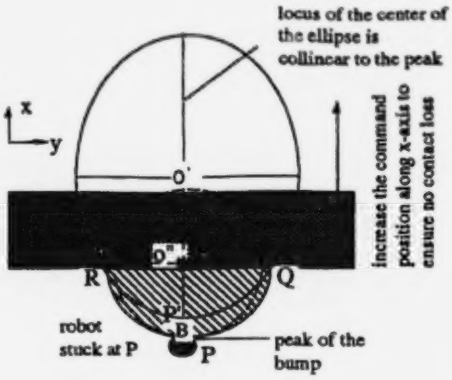

locus of the center of the ellipse is collinear to the peak
x
y
O'
O"
R
Q
P'
B
P
robot stuck at P
peak of the bump
increase the command position along x-axis to ensure no contact loss

Fig. 8 The robot is stuck at the peak when ellipse center is collinear to the peak

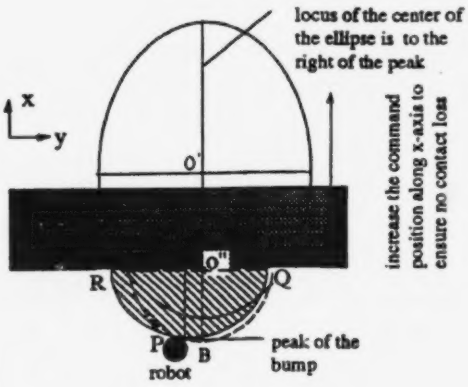

locus of the center of the ellipse is to the right of the peak
x
y
O'
O"
R
Q
P
B
robot
peak of the bump
increase the command position along x-axis to ensure no contact loss

Fig.9 The locus of the center of the ellipse is to the right of the peak

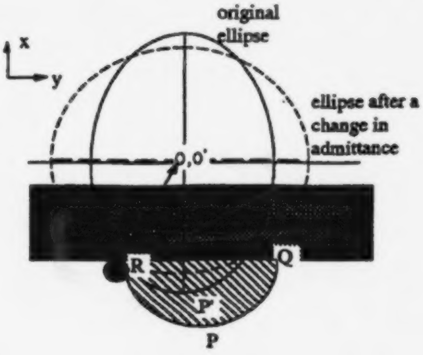

original ellipse
x
y
ellipse after a change in admittance
O,O'
R
Q
P'
P

Fig. 10 The regulation of admittance shown as a change in the shape of the ellipse

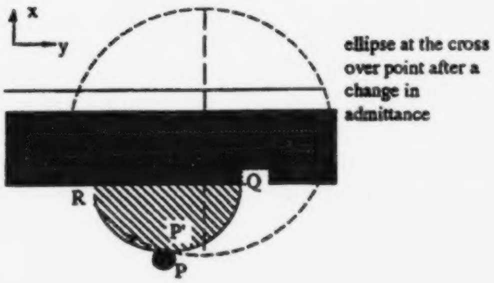

x
y
ellipse at the cross over point after a change in admittance
R
Q
P'
P

Fig. 11 The ellipse at the bump cross over point after appropriate regulations in the admittances

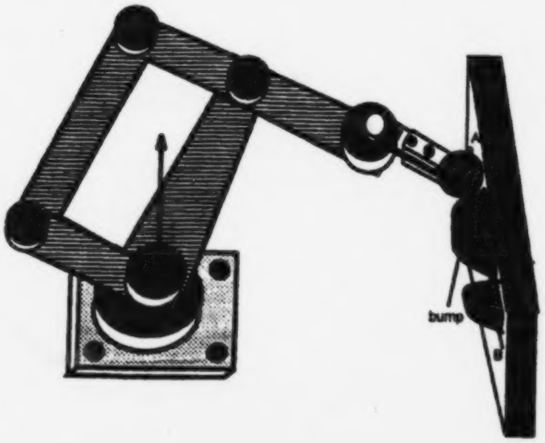
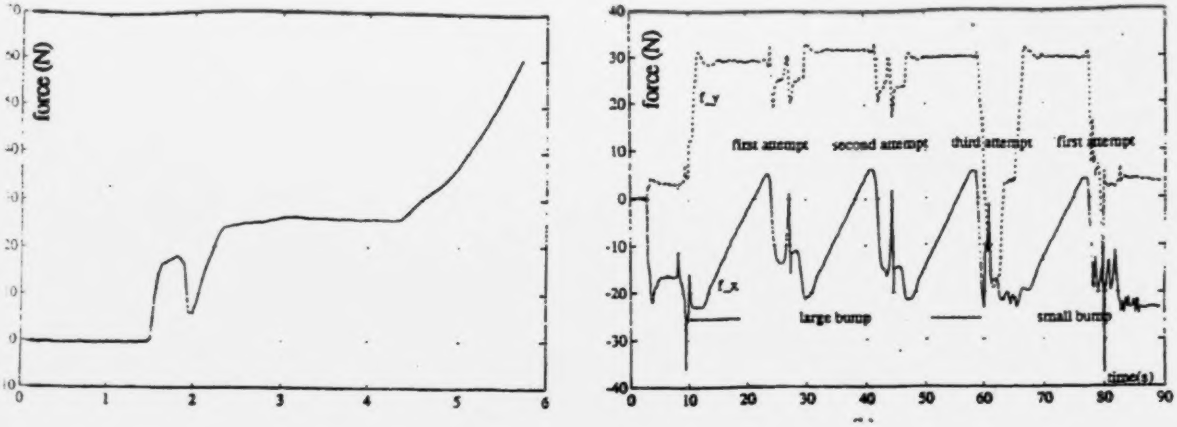
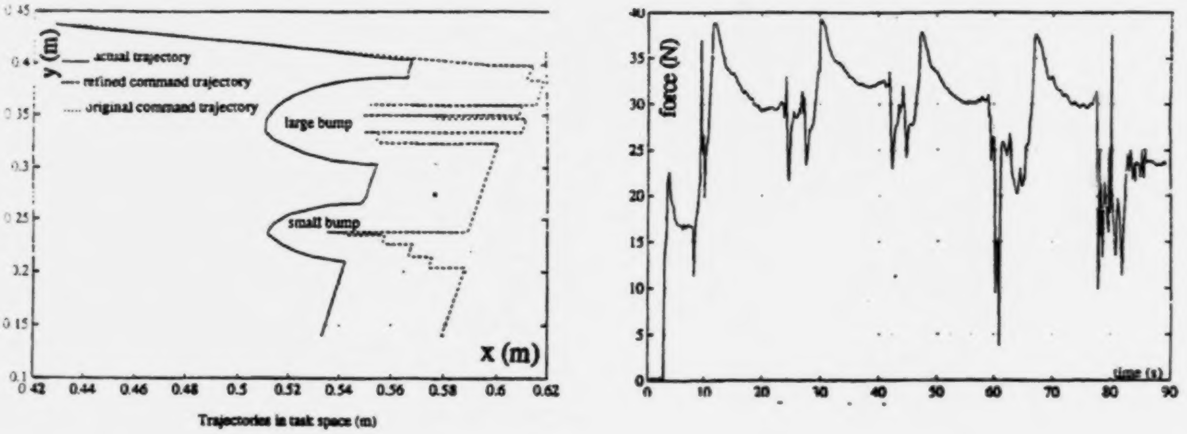


Fig. 12 An overview showing two different sizes of bumps along the trajectory of the robot



(a) (b)

Fig. 13 (a) Contact force along Y-axis when there is no qualitative supervisory control (b) Contact force along the X and Y axis with a qualitative supervisor



(a) (b)

Fig. 14 (a) Contact trajectories (refinements) in task space (b) Resultant contact force at the tip of the manipulator

178920

Reinforcement Learning and The Parametric Motor Control Hypothesis Applied to Robotic Arm Movements

J. Lipitkas
Institute for Space and Terrestrial Science
and, Institute for Aerospace Studies, University of Toronto,
4925 Dufferin St., Toronto, ON, Canada, M3H 5T6.

G.M.T. D'Eleuterio
Institute for Aerospace Studies, University of Toronto,
4925 Dufferin St., Toronto, ON, Canada, M3H 5T6.

O. Bock
Institute for Space and Terrestrial Science, and York University,
4850 Keele St., Toronto, ON, Canada, M3J 1P3.

J.J. Grodski
Defence and Civil Institute of Environmental Medicine,
1133 Sheppard Ave. West, Toronto, ON, Canada, M3M 3B9.

#149629

Abstract

In a previous study we presented a novel control scheme for point-to-point movements of robotic manipulators where explicit calculations of torque-time trajectories are no longer required. Signals specifying the initial and desired final configuration of the manipulator are mapped by an artificial neural network (ANN) into signals representing scaling parameters; the latter are used to scale prototypical joint-torque functions which serve as torque reference signals for the robot's actuators. Therefore, only a small number of parameters need to be determined for each movement instead of the entire torque-time trajectory. The appropriate mapping of the initial and desired final configuration into scaling parameters was learned by training the ANN. In classical training procedures, various inputs are applied to the ANN, the actual ANN outputs are compared to the desired ones, and the weights are updated to minimize the difference between the actual and desired outputs. However, applying these procedures to robot control is especially troublesome since the desired performance is defined in terms of robot kinematics (here: final gripper position) rather than in terms of ANN outputs (here: scaling parameters). In contrast to the classical training procedures, Reinforcement Learning does not calculate the difference between actual and desired ANN output and thus is potentially immune to this problem. Essentially, the foundation of this approach is the use of position and velocity sensor information directly to produce the scaling parameters, and is based on the idea that the tendency to produce an action is strengthened if that action previously led to consequences that have been considered desirable.

Hence, this paper expands on the previous work by presenting a reinforcement based ANN system that identifies the relationship between the task requirements and the torque parameters for controlling movements of robotic manipulators. The use of reinforcement learning allows the robot arm to adapt interactively to the environment on an ongoing bases without requiring an explicit system model, and eliminates the need for a priori knowledge of the desired ANN outputs. Thus, our control concept is able to achieve an accurate terminal approach to the targets by the addition of sensory feedback, as found in humans. This control concept is particularly useful for real-time robot control in unstructured environments because of the ongoing interaction between the robot and the environment. In simulation studies, this control concept discriminated between targets, and generalized to unlearned target positions with-

out the need to specify the end-effector trajectory since the trajectory is part of the solution.

Introduction

An important problem in robotics is the generation of motor commands to move the end-effector from its initial position into a desired final position. Recent approaches to solve this problem have included the use of artificial neural networks (ANNs) to map, for each point in time, the values of kinematic variables into the corresponding joint torque values [5], [7], [10], [11]. Unfortunately, the explicit calculation of joint torque profiles by ANNs is computationally very intensive, and therefore cumbersome for applications where torques must be calculated in real-time.

We have recently introduced an approach, called *Parametric Control* which disposes of the need to calculate torque profiles explicitly as outlined in [8] and [3]. As shown in Figure 1, signals specifying desired final gripper position (x_d and y_d in Cartesian coordinates) and the actual initial manipulator configuration (joint angle functions $cos\,\theta_1$, $sin\,\theta_1$, $cos\,\theta_2$, and $sin\,\theta_2$) are sampled only once before each movement as ANN inputs. ANN outputs are scaling parameters (p_i, three per joint) that scale the sinusoidal signals provided by a function generator: the parameters define the magnitude of first and the second half-wave as well as the time of cross-over, with total duration set at 4 s. The scaled time-functions serve as torque reference signals to the robot's actuators. We have documented [8], [3] that Parametric Motor Control allows the accurate execution of point-to-point gripper movements without the need for real-time processing, once the ANN is trained to map desired and actual positions appropriately into scaling parameters.

Various ANN training procedures are available in the literature. Typically different inputs are applied to the ANN, the actual outputs are compared to the appropriate ones, and ANN weights are then adjusted to reduce the difference between actual and appropriate outputs. However, a difficulty arising in robotic applications is that the appropriate ANN output is often unknown, since the desired performance is defined in terms of gripper kinematics rather than in terms of ANN output. Proposed solutions to this *Distal Error Problem* [7] involve the use of additional ANNs and thus substantially increase the computational complexity. We have used one of these solutions, called *Direct Inverse Modeling*, in our previous study.

The purpose of the present work is to implement an alternative training procedure called Reinforcement Learning ([1], [12]) which allows the formulation of performance criteria in reference frames other than ANN output, and hence is potentially immune to the distal error problem. Reinforcement Learning is based on the idea that the tendency to produce an action is strengthened if the consequences of that actions have previously been considered desirable ([1], [12]). Original formulations have been limited to the selection between discrete actions. Recently, however, the method has been expanded for continuous signals ([6]). We adopted this expanded version for our Parametric Motor control approach. Our results indicate that Reinforcement Learning is faster and more expedient than Direct Inverse Modeling, and offers several potential benefits for robot control.

Reinforcement Algorithm

An important feature of the Reinforcement Learning algorithm is the use of stochastic behaviour in the form of a computational unit with a random component. This random component, or noise, is used to prevent a search process from converging to a local minimum ([2]). The network processing units (i.e. neurons) compute a random output according to a standard normal distribution with mean, μ, and standard deviation, σ: μ is an estimate of the optimal output, and σ controls the area of the search around which the optimal output might be found. The first step is to determine the value of these parameters.

The mean of the processing units, μ, is computed as the weighted sum of the inputs to that unit, i.e.

$$\mu(t) = \sum_{i=1}^{n} \omega_i x_i(t) \qquad (1)$$

where the parameters ω_i denotes an element of the weight vector consisting of all the weights of the unit's input, x_i denotes an element of the vector of input signals, and t denotes a training example.

The standard deviation, σ, should be small if the expected outputs of the network are close to optimal, and it should be large otherwise. To this end, it has been shown that ([6])

$$\sigma(t) = max\Big(k_\sigma(1.0 - \hat{r}(t)), 0.0\Big) \qquad (2)$$

where k_σ is a constant and $\hat{r}(t)$ is the expected reinforcement. In the simulations presented in this paper, $k_\sigma = 0.2$.

The processing units compute their stochastic outputs by using the mean, μ, and the standard deviation, σ, parameters to generate a normally distributed random activation level $a(t)$, i.e.

$$a(t) = N(\mu(t), \sigma(t)). \tag{3}$$

The activation $a(t)$ is then transformed into the output of the unit $y(t)$ by using the sigmoidal function,

$$y(t) = \frac{2.0}{1.0 + e^{-a(t)}} - 1.0. \tag{4}$$

The weights are modified according to

$$\Delta\omega(t) = \alpha\Big(r(t) - \hat{r}(t)\Big)e(t) \tag{5}$$

where α is a leaning rate, $r(t)$ is the actual reinforcement signal received from the environment, $\hat{r}(t)$ is the expected (i.e. predicted) reinforcement, and e is the *characteristic eligibility* of the weights ω [13] (a complete description of the actual reinforcemtn siganl will be presented in the next section).

A number of alternatives exist for representing the expected reinforcement (for example see [9]), but in this paper, a *predictive comparison* [13] technique will be used. This technique computes $\hat{r}(t)$ based on the environmental reinforcement received when the input stimulus x_i is present. In order to undertake this task, a critic is built as a second set of weights of the network i.e.

$$\hat{r}(t) = \sum_{i=1}^{n} v_i(t)x_i(t) \tag{6}$$

where v is the weight vector used to calculate the expected reinforcement.

For updating v_i, the following general equations are used ([6])

$$v_i(t+1) = v_i(t) + \beta\Delta v(t)x_i(t) \tag{7}$$

where β is a learning rate parameter and

$$\Delta v(t) = r(t) - \hat{r}(t). \tag{8}$$

We have found that using $\beta > \alpha$ improves the learning process. Typical values for the learning rates used were, $0.001 < \alpha < 0.03$ and $0.008 < \beta < 0.08$.

A number of different eligibility factors exist [9]. The *discrepancy-based* mechanism which evaluates the difference between the actual output and the expected one as proposed in [6] was used in the simulations presented in this paper, therefore

$$e(t) = y(t)\frac{a(t) - \mu(t)}{\sigma(t)}. \tag{9}$$

Reinforcement Schedule

In reinforcement learning algorithms, an actual and a desired configuration are compared, and a reinforcement signal based on the error between the two configurations is sent to the ANN. In Parametric Motor Control, ANN performance is considered satisfactory when the network output values p_i, applied to the function generator, result in a gripper movement, such that the actual and desired final positions are the similar, i.e. $x_a \approx x_d$ and $y_a \approx y_d$, and the velocity of the end-effector is zero at the desired final position. In our simulations, the reinforcement signal $r(t)$ for each training example t is defined as

$$r(t) = (1 - e_3)e_3 + e_3e_4 \tag{10}$$

e_3 is a function of the end-effector position error and is defined as

$$e_3 = \frac{1}{1 + e_1} \tag{11}$$

where

$$e_1 = |x_d - x_a| + |y_d - y_a|. \tag{12}$$

Correspondingly, e_4 is a function of the end-effector velocity error defined as

$$e_4 = \frac{1}{3e_1 + e^{\frac{e_2^2}{e_1}}} \tag{13}$$

where e_2 is the tangential end-effector velocity at the end of the movement and is defined as

$$e_2 = |V_x| + |V_y| \tag{14}$$

where V_x and V_y are the velocities of the end-effector in the x and y directions respectively.

The reinforcement signal $r(t)$, as specified above, guarantees that as the robot end-effector approaches it's desired final position (i.e. $e_1 \rightarrow 0.0$), the velocity of the end-effector will predominate the reinforcement signal, i.e.,

$$\lim_{e_1 \rightarrow 0} r(t) \sim f(e_4) \tag{15}$$

Furthermore, at the start of training when $e_1 \gg 0$, the position error of the end-effector will predominate the reinforcement signal, i.e.,

$$\lim_{e_1 \gg 0} r(t) \sim f(e_3). \tag{16}$$

Results

We evaluated the performance of our approach by using a simulation of a two joint planar robotic arm called Radius [4] working in a 5 × 3 target array in a 80 × 30 cm (horizontal × vertical) workspace as illustrated in Figure 2. The end-effector was positioned at one of the targets, thus specifying θ_a, and a different target was selected as desired final position, i.e., x_d, y_d. The final errors of the resulting movement were noted, and a new trial using a different x_d, y_d and θ_a was initiated. This procedure was carried out for a total of 225 possible trials.

ANN performance before training and after 3000 iterations of the Reinforcement Learning algorithm are shown in Table 1. Clearly, the errors between the desired and actual final end effector positions are greatly reduced, dropping from an average error of 0.75 m to 0.0567 m. As a comparison, the inter-target distances were 20 cm in the horizontal direction, and 10 cm in the vertical direction. Table 1 also documents the performance of the trained network with a set of targets that were not used for training (i.e., untrained). These targets were yielded by shifting the original target set by one half of the inter-target distance horizontally and vertically, and thus represent the *most different* locations that could be obtained in the workspace. The errors for the untrained targets averaged 0.0656 m. Thus, they were slightly larger than those obtained with the original target set.

Conclusions

We have described a solution to the control of point-to-point movements of a two joint planar robotic arm using ANNs and reinforcement learning, applying it to the Parametric Motor Control hypothesis first proposed in [3] and [8]. The results of this study substantiate the effectiveness of the motor program hypothesis in robotic control. In addition, the use of reinforcement learning allowed the robot arm to adaptively interact with the environment on an ongoing bases without requiring explicit knowledge of manipulator dynamics, and eliminated the need for a priori knowledge of the desired ANN outputs. Furthermore, we have documented that the trained network discriminated between targets and generalized to unlearned target locations without the need to specify the end-effector trajectory since the trajectory was part of the solution.

Although results have been presented for a two-link robotic arm, the control procedure can in principle be extended to systems with an arbitrary number of degrees-of-freedom. Finally, it is significant to note that this concept owing to its use of prototypical torque profiles can also be extended to incorporate varying-speed capabilities and load-compensation strategies.

Acknowledgments

We gratefully acknowledge the financial support provided by the DCIEM under contract W7711-0-7118. Further support was provided by the Institute for Space and Terrestrial Science, and the University of Toronto Institute for Aerospace Studies.

References

[1] Barto A.G., Sutton R.S., Anderson C.W. (1983), "Neuronlike Adaptive Elements That Can Solve Difficult Learning Control Problems," *IEEE Trans on Systems, Man, and Cyber*, Vol. 13, No. 5, 834-846.

[2] Barto A.G. (1985), "Learning by Statistical Cooperation of Self-Interested Neuron-Like Computing Elements," *Human Neurobiology 4*, 229-256.

[3] Bock O., D'Eleuterio G.M.T., Lipitkas J., Grodski J.J (1993), "Parametric Motion Control of Robotic Arms: A Biologically Based Approach Using Neural Networks," Proc. of the 1993 Goddard Conference on Space Applications of Artificial Intelligence.

[4] Carusone J., D'Eleuterio G.M.T. (1991), "Experiments in the control of structurally flexible manipulators with the Radius facility," Proc. of the Second Joint Japan/U.S. Conference on Adaptive Structures, Nagoya, Japan, 589-605.

[5] Graham D.P.W., D'Eleuterio G.M.T. (1991), "MOVE - A Neural Network Paradigm for Robotic Control," *CASI Journal* Vol 37, No. 1, 17-26. .

[6] Gullapalli V. (1990),"A Stochastic reinforcement learning algorithm for learning real-valued functions," *Neural Networks*, Vol. 3, 671-692.

[7] Jordan M.I., Rumelhart D.E. (1991), "Forward models: Supervised learning with a distal teacher," Occasional Paper No. 40, Center for Cognitive Science, MIT, Cambridge, MA.

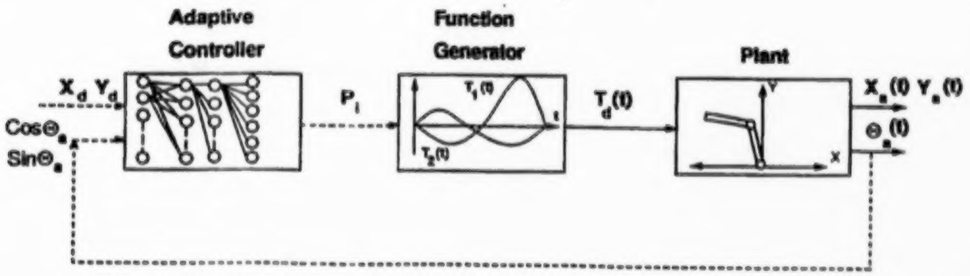


Figure 1: *Overview of control concept. Dashed lines indicate signals used only once before each movement; solid lines indicate signals used continuously during movement (X_d, X_a, Y_d, Y_a - desired and actual end-effector postiions, $Cos\theta_a, Sin\theta_a$ - the actual cosine and sine of the initial joint angular positions, $\theta_a(t)$ - actual joint angular positions, $T_i(t)$ - joint torques, P_i - parameters used to describe the joint torques).*

	Average Final Position Error (m)
Before training	0.75
After training on trained data set	0.0567
Untrained data set	0.0656

Table 1: *Table of average final position errors from the actual to the desired final end-effector positions.*

[8] Lipitkas J., Bock O., D'Eleuterio G.M.T., Grodski J.J. (1992), "Parametric Control of Point-to-Point Robotic Arm Movements Using Artificial Neural Networks," Proc. of the Seventh CASI Conference on Astronautics, Ottawa, Canada, 167-173.

[9] Millan J. Del R., Torras C. (1992), "A Reinforcement Connectionist Approach to Robot Path Planning in Non-Maze-Like Environments," *Machine Learning* 8, 363-395.

[10] Miller W.T. (1989), "Real-Time Applications of Neural Networks for Sensor-Based Control of Robots with Vision," *IEEE Trans on Systems, Man, and Cyber*, Vol. 19, No. 4, 825-831.

[11] Miyamoto H., Kawato M., Setoyama T., Suzuki R. (1988), "Feedback-Error-Learning Neural Network for Trajectory Control of a Robotic Manipulator," *Neural Networks*, Vol. 1, 251-265.

[12] Sutton R.S., Barto A.G., Williams R.J. (1991), "Reinforcement Learning is Direct Adaptive Optimal Control," Proc. of the American Control Conference, Boston, MA., USA, 2143-2146.

[13] Williams R.J. (1992), "Simple Gradient-Following Algorithms for Connectionist Reinforcement Learning," *Machine Learning* 8, 229-256.

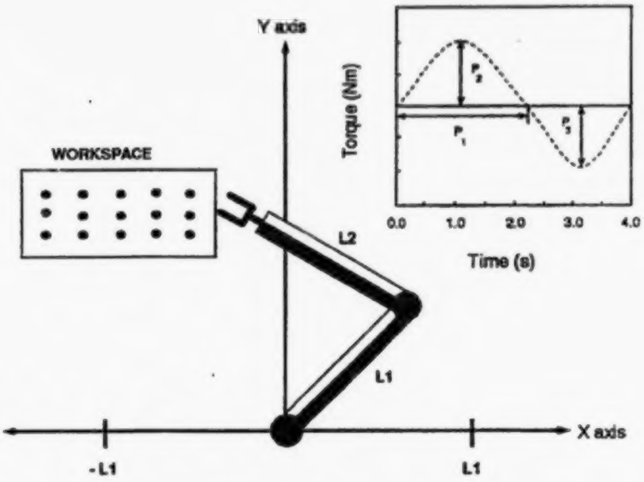

Y axis
Torque (Nm)
P_2
P_1
P_3
0.0
1.0
2.0
3.0
4.0
Time (s)
WORKSPACE
L2
L1
X axis
- L1
L1

Figure 2: *Illustration of the manipulator and it's workspace.*

223600

Integrated Mechanical and Control Design For Well–designed Flexible Joint Robots

Hoda A. ElMaraghy & Feng Liang
Flexible Manufacturing Center
McMaster University
Hamilton, Ontario Canada L8S 4L7

#149631

Abstract

The idea of integrated mechanical and control design for well–designed flexible joint robots is proposed in this paper. It provides an approach for designing simpler control laws which take both mechanical parameters and control parameters into consideration. Designing flexible joint robots with variable structures can ensure that the robots possess some adjustable mechanical parameters, which can be utilized as their controller parameters for different tasks. The paper shows how the controller design of flexible joint robots can benefit from the integrated design, and defines the concept of well–designed robots. As a result, simple partial inverse control laws are derived for the well–designed flexible joint robots. Some constraints on the masses, lengths, shapes, or joint stiffness of the robots are imposed for them to be well–designed. Passive damping design of the robot link dynamics can make the robots possess some desired impedance, so destructive collision can be avoided without designing force controllers, when the robots contact the environment. In these control laws, the parameters of the robot link dynamics affect the trajectory control accuracy only, but not the stability of the robotic control systems. When the parameters of the robotic systems are unknown, the given adaptive and learning scheme provides a feasible remedy for the above control laws. Simulation results verify the feasibility of the new control schemes. The research on well–designed flexible joint robots and variable structure robots can provide us with deeper insight into robotic systems design.

Introduction

A general trend in robot manipulator design is to make them light relative to their loads, working at higher speed with high precision, adaptable to different tasks, cost–effective and efficient in applications. At present, there are both special and general purpose robots, which were intensively studied in the past decades. The presently available robots are initially designed with fixed structures, then various controllers are designed subsequently based on these fixed robotic dynamics. This procedure often leads to complex control laws with low robustness to model uncertainty and measurement noise. Innovations are needed to make the current robots fit the special demands of different applications.

Integrated mechanical and control design is a new approach to robotic system design. It combines the mechanical design and control design, which were done separately for most existing systems, together to simplify and optimize both of them. The robotic dynamics are regarded as adjustable by changing mechanical structures or parameters, such as the masses, lengths, shapes, joint stiffness, connectivity relations, and are used to improve the control performance. As an example, the paper shows how the controller design of flexible joint robots can benefit from this integrated design approach. The concept of *well–designed flexible joint robots* is defined for such robots whose link dynamics possess some special characteristics (e.g., stability and fast response). Simple partial inverse control laws are derived for the well–designed flexible joint robots, where the robot link dynamics becomes the *internal dynamics* of the control systems and their parameters affect the trajectory control accuracy only, but not the stability of the robotic control systems. It shows as well that adding mechanical dampers to the joints can make the system dynamics "dissipative", and therefore, simplify the controller design. Joint flexibility is not a demerit using this approach, but rather an advantage for compliant robotic control.

As for the control of flexible joint robots, there are many published studies. The singular perturbation approach is first proposed to solve the joint flexibility problem [1–4]. Another research direction is the adaptive control approach [5–8], which allows some uncertainty in system model parameters. The invariant manifolds approach and nonlinear transformation approach [9–11] give similar results as the feedback linearization or inverse dynamics approach [12–14] do, when they are applied to the robotic control problem. These approaches usually require accurate knowledge of the system parameters or measurements of high–order derivatives of joint angles. There are also robust control approaches [15–18] and learning control schemes [19] to deal with this problem. Compared with existing schemes, the one presented in this paper focuses on deriving simple control laws for the flexible joint robots, since simplicity of a control algorithm is very important for real–time implementation and acceptance by application engineers. The current adaptive control schemes and robust control schemes are more complex and expensive than conventional control schemes, in spite of the availability of inexpensive microcomputers for implementation. It is worth mentioning that the research progress in the cascaded nonlinear systems [27–28] and the research on the dissipative nonlinear systems [29] will provide solid theoretical support for the concept of well–designed flexible joint robots.

The paper is organized as follows. Section 2 introduces the integrated mechanical and control design concept, and illustrates the idea of introducing variable structure robots to facilitate the integrated mechanical and control design under different control tasks. Section 3 applies the new concepts to the control of well–designed flexible joint robots. Simple partial inverse control laws are derived. Section 4 gives adaptive and learning modifications of the control laws when the parameters of the flexible joint robots are highly uncertain. Section 5 presents the simulation results of trajectory control of two–link planar robots with flexible joints and shows the feasibility of the new control laws. Section 6 provides conclusions.

The Concept of Integrated Mechanical and Control Design

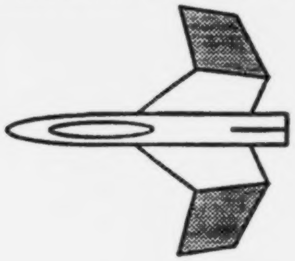
Searching for simple and effective control laws which are less computation–intensive is important for control engineers. Recalling the development history of control systems design, we observe that the designs of most control systems take place after their mechanical designs are finished. That is, the mechanical design of a dynamical system is done mainly.according to its functions and mechanical practicability without considering the design of the control system. Many design degree of freedoms are not used and the corresponding parameters are determined by experience. In general, control was considered as a means of improving the mechanical performance after the fact. There is no active interaction between the mechanical design and control system design so as to simplify and optimize both designs. As a result, the designs of many dynamical systems are not globally optimal, and designers often encounter two problems.

The first problem, encountered by mechanical engineers, is that many mechanical systems must be made huge or complex, with some special mechanical configurations [20–21], to guarantee system dynamics decoupling, stability or linearity. This approach to mechanical system design is considered reasonable since it is difficult to design and implement a sophisticated control law for a mechanical system with complex dynamics. However, it is not the best solution for simplifying control system design.

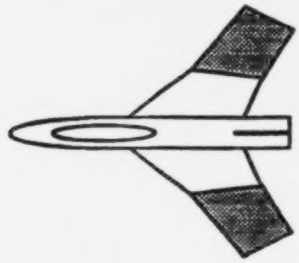
The second problem, encountered by control engineers, is that as the system performance requirements increase, dynamical systems become complex, highly coupled and nonlinear with uncertain or time–varying parameters. Robots, aircraft, space flying vehicles and weapon systems are examples of such control systems. Designing control systems for these dynamical systems is hard work, and usually results in very complex control laws. Furthermore, the control theory for general nonlinear systems is still under development, and many applications of control systems are not easy to accomplish at present.

It is obvious that the performance of a dynamical system is determined by both its mechanical design and control system design. The fact that each of them may be individually optimal does not guarantee that the overall dynamic system is optimal. If the mechanical design and control system design are integrated together, better and simpler design of dynamical systems may be achieved. This gives rise to the concept of integrated mechanical and control design, so as to find simple and effective control laws which are less computation–intensive, and simple mechanical designs. Therefore, it is not necessary to consider system dynamical models derived from mechanical design as fixed, and the mechanical parameters, normally defined by functional and physical constraints, can be determined or optimized taking into consideration the requirements of control design as well.

For special purpose robots, good system performance can be achieved by integrated mechanical and control design. Then the mechanical structures and controllers are fixed. For general purpose robots, different tasks may require different mechanical parameters for optimal system performance. Forcing the robots to have variable structures ensures that the robots possess some variable mechanical parameters. Such parameters can be utilized as the controller parameters to adjust control system performance for different control tasks. This type of robots is called the variable structure robots.



Configuration 1



Configuration 2

Fig. 1 The advanced variable structure fighters

Variable structure robots provide more freedoms for different control task designs. This concept is relatively new, although variable structure fighters in aeronautics are already a reality. Fig. 1 shows two different configurations of an advanced fighter. By changing the wing shapes, different flight dynamics are obtained, and optimal lift forces can be achieved at different altitudes and air speeds.

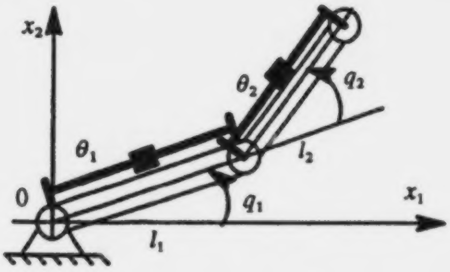


Fig. 2 Variable structure robots with movable mass blocks

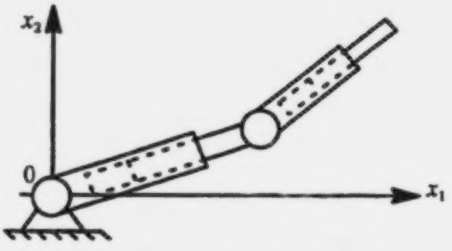


Fig. 3 Variable structure robots with variable length links

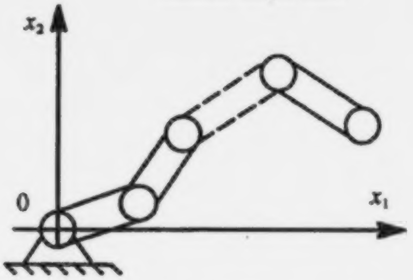


Fig. 4 Variable structure robots with modular redundant links

There are many ways to achieve variable robot structures. For example, Fig. 2 shows a two–link planar robot with movable mass blocks. Changing the positions θ_i of the mass blocks can change the dynamics of the robots, such as the inertia matrix. Fig. 3 shows a two–link planar robot with variable length links. Changing the lengths of the links changes the dynamics of the robots to achieve minimum energy control. For instance, if the robot is required to move from point *A* to point *B*, the robot first shrinks its link lengths to become compact, then retains its compact configuration when approaching point *B*. When near point *B*, it adjusts its link lengths to reach point *B*. Therefore, minimum energy control can be achieved using this strategy. Fig. 4 shows a planar variable structure robot with modular redundant links. Only two joints are allowed to move at an instant. Fixing different joint connections results in different modular robots with different dynamics for different control tasks or different phases in one control task. There may be other schemes, e.g., robots designed for dynamic passivity.

In general, *n*–link variable structure robot manipulators with flexible joints and free end–effector can be expressed as:

$$M(q,\theta)\ddot{q} + H(q,\dot{q},\theta,\dot{\theta}) + K(q-p) = 0 \qquad (1)$$

$$I_m\ddot{p} + B_m\dot{p} + F_m - K(q-p) = u \qquad (2)$$

where $q = [q_1, ..., q_n]^T$ is the vector of joint coordinates; θ is the adjustable mechanical parameter vector; $M(q,\theta)$ is the positive definite inertial matrix; $H(q,\dot{q},\theta,\dot{\theta})$ is the vector of centrifugal, Coriolis and gravitational term; $p = [p_1, ..., p_n]^T$ is the vector of rotational angles of actuator rotors; $I_m = diag[I_1, ..., I_n]$, where I_i is the moment of inertia of the *i*th rotor; $B_m = diag[B_{m1}, ..., B_{mn}]$, where B_{mi} is the viscous friction coefficient of the *i*th joint; $F_m = [F_{m1}, ..., F_{mn}]^T$, where F_{mi} is the friction torque acting on the *i*th rotor; $K = diag[K_1, ..., K_n]$, where K_i is the equivalent elastic constant of the *i*th joint.

If the variable parts of the robots can also be controlled, then their dynamics can be expressed as:

$$N(q,\theta)\ddot{\theta} + G(q,\dot{q},\theta,\dot{\theta}) = \tau \qquad (3)$$

where τ is the control vector of the variable parts of the robots. Thus more control inputs are introduced to the system to achieve better control performance.

In the next section, the above concepts are applied to the control of flexible joint robots (FJR). As a result of active coordination between control system design and mechanical design, some mild constraints on both controller parameters and mechanical parameters are defined to make a robot well–designed.

Control of Well–Designed FJR

Let us examine the link dynamics Eq. (1) of variable structure robots. Assume that θ will be determined by the integrated design and remain constant for a specific given control task. Then $H(q,\dot{q},\theta,\dot{\theta})$ in Eq. (1) can be decomposed as follows:

$$H(q,\dot{q},\theta,\dot{\theta}) = C(q,\dot{q},\theta)\dot{q} + g(q,\theta) \qquad (4)$$

where the first term represents the centrifugal and Coriolis moment vector; the second term represents the gravitational moment vector. Substituting Eq. (4) into Eq. (1) and reorganizing the terms, we have:

$$M(q,\theta)\ddot{q} + C(q,\dot{q},\theta)\dot{q} + g(q,\theta) + Kq = Kp \qquad (5)$$

If we take the joint angle vector q as the output vector directly, as done by many researchers, complex high order nonlinear equations have to be solved to obtain feedback linearization controllers. However, if we take the motor angle vector p as the output vector, then Eq. (2) is the input–output representation of the flexible joint robots with their *internal dynamics* [22] as Eq. (1) or (5). It can be seen that the left–hand side of Eq. (5) is the link dynamics which is only related to the joint angular vector q, $\dot{q}$ and $\ddot{q}$, and the right–hand side term Kp can be regarded as the external forcing input vector. The *zero dynamics* w.r.t. output vector p is:

$$M(q,\theta)\ddot{q} + C(q,\dot{q},\theta)\dot{q} + g(q,\theta) + Kq = 0 \qquad (6)$$

It can be seen that Eq. (6) is totally determined by the mechanical design of the robot, i.e., the masses, lengths, shapes (moment of inertia), joint stiffness, connectivity relations and link materials. The *zero dynamics* is an inherent characteristic of a dynamic system with selected outputs. It is shown in [23] that the solution q=0 and $\dot{q}$=0 of Eq. (6) is globally stable in the sense of Liapunov, i.e., q and $\dot{q}$ are uniformly bounded under bounded initial disturbances. Therefore, the system internal dynamics is bounded–input bounded–output (BIBO). This dynamic feature will be utilized to design simpler robot controllers. In summary, by changing the definition of the output vector, we can simplify the high order nonlinear system control design problems into reduced order nonlinear system control design problems.

Assume that the desired trajectory of the end–effector of a robot manipulator is expressed in joint coordinates as q_d, $\dot{q}_d$, $\ddot{q}_d$ and the desired trajectory of the rotational angles of actuator rotors is expressed as p_d, $\dot{p}_d$, $\ddot{p}_d$, which is depended on q_d, $\dot{q}_d$, $\ddot{q}_d$. Let

$$u = I_m(\ddot{p}_d - D_1\dot{\delta} - D_2\delta) + B_m\dot{p} + F_m - K(q - p) \qquad (7)$$

where

$$\delta = p - p_d\,, \qquad \dot{\delta} = \dot{p} - \dot{p}_d \qquad (8)$$

$$p_d = K^{-1}[M(q_d,\theta)\ddot{q}_d + C(q_d,\dot{q}_d,\theta)\dot{q}_d + g(q_d,\theta)] + q_d \qquad (9)$$

$$\dot{p}_d = \frac{dp_d}{dt} \qquad \ddot{p}_d = \frac{d^2p_d}{dt^2} \qquad (10)$$

and D_1 and D_2 determine the convergence speed of $\| \delta \|$. Substituting (7) into (2) gives:

$$\ddot{\delta} + D_1\,\dot{\delta} + D_2\,\delta = 0 \qquad (11)$$

Obviously, we can always make $\| \delta \| \rightarrow 0$ by properly choosing D_1 and D_2 according to the desired performance. Therefore, control law (7) can make the motor angles of flexible joint robots track the desired trajectory p_d, $\dot{p}_d$, $\ddot{p}_d$ asymptotically, and at the same time, the joint angle q determined by Eq. (12) is bounded, due to the dynamic characteristics proved in [23].

$$M(q,\theta)\ddot{q} + C(q,\dot{q},\theta)\dot{q} + g(q,\theta) + Kq = K\delta + Kp_d \qquad (12)$$

Therefore, eqs. (11) and (12) consist of a BIBO system. Compared with the existing control schemes for flexible joint robots, this one is among the simplest. If we can further make $q(t) \rightarrow q_d(t)$, then the trajectory control problem of flexible joint robots is solved.

Definition 3.1: If the global stable zero dynamics (6) of a flexible joint robot possesses some desired characteristic (e.g., well–damped, fast response) which guarantees that control law (7)–(10) can also make the robotic system (1) and (2) track the desired trajectory q_d, $\dot{q}_d$, $\ddot{q}_d$ with desired performance, then the flexible joint robot is called mechanically *well–designed.*

To make a flexible joint robot well–designed, the mechanical parameters of Eq. (6) have to satisfy some constraints for selected D_1 and D_2. From (5), (8) and (9), we get:

$$M(q,\theta)\ddot{q} + H(q,\dot{q},\theta) + Kq - [M(q_d,\theta)\ddot{q}_d + H(q_d,\dot{q}_d,\theta) + Kq_d] = K\delta \qquad (13)$$

Defining $e(t)$=$q(t) - q_d(t)$ and linearizing Eq. (13) along the desired trajectory $q_d(t)$ gives:

$$M(q_d,\theta)\ddot{e} + A_1(q_d,\theta)\dot{e} + [A_2(q_d,\theta) + K]e = K\delta \qquad (14)$$

where

$$A_1(q_d,\theta) = \left[\frac{\partial H}{\partial \dot{q}}\right]^T_{\substack{q=q_d\\ \dot{q}=\dot{q}_d}} \quad A_2(q_d,\theta) = \left[\frac{\partial H}{\partial q}\right]^T_{\substack{q=q_d\\ \dot{q}=\dot{q}_d}} \qquad (15)$$

Thus eqs. (11) and (14) define the tracking error dynamics of the robotic control system under control law (7)–(10). Since they are linear time–varying, we know, qualitatively, that changing the parameters in $M(q,\theta)$, $H(q,\dot{q},\theta)$, K, D_1 and D_2 can make $e(t) \rightarrow 0$ at fast speed, that is, make the end–point of the controlled robot track the desired trajectory, provided that the system model is fairly accurate and the initial trajectory errors are bounded. The coupling between eqs. (11) and (14) can introduce damping effects to link dynamics. Thus, simply changing the mechanical structure parameters, such as the masses, lengths, shapes, joint stiffness, connectivity relations, can make a flexible joint robot well–designed with given desired performance. The introduction of variable structures to robots provides more adjustable parameters θ to make the robots well–designed. This will be illustrated by simulation.

Since Eq. (14) is linearized from a nonlinear equation, only local asymptotical tracking performance of the robotic control system can be theoretically proved. However, the region of local asymptotical stability is fairly large, as shown by simulations, if the system model is fairly accurate. If the system model is of large error, the joint angle tracking errors can always be confined to be bounded since the system described by eqs. (11) and (12) is globally BIBO. To achieve high tracking accuracy in this case, the adaptive and learning modifications of the control law (7)–(10), given in next section, can be utilized.

It is worth indicating that the above approach is different from the singular perturbation approach [1–3], in which the assumption of separable fast and slow system dynamics is needed. Here, the global stability of the internal dynamics of flexible joint robots is utilized to achieve simplified input–output feedback linearization. Therefore, separation of system dynamics is not required, and high joint flexibility is allowed.

Definition 3.2: Assume that the generalized force vector $Q(q,\dot{q})$ is exerted on a mechanical system. If there exists a continuous, strictly increasing function $a(.)$ with $a(0)$=0 such that, for every q and $\dot{q}$: $Q^T(q,\dot{q})\dot{q} \leq -a(\| \dot{q} \|)$, then $Q(q,\dot{q})$ is called complete dissipative generalized forces [23]. Viscous friction force is an example of complete dissipative generalized forces.

If complete dissipative torques $Q(q,\dot{q})$ are exerted on the robot links, then Eq. (5) becomes:

$$M(q,\theta)\ddot{q} + C(q,\dot{q},\theta)\dot{q} + g(q,\theta) + Kq = Q(q,\dot{q}) + Kp \tag{16}$$

and the corresponding zero dynamics is:

$$M(q,\theta)\ddot{q} + C(q,\dot{q},\theta)\dot{q} + g(q,\theta) + Kq = Q(q,\dot{q}) \tag{17}$$

From [32], we know that q=0 and $\dot{q}$=0 is a globally asymptotical stable solution of equation (17). If (9) is modified as follows:

$$p_d = K^{-1}[M(q_d,\theta)\ddot{q}_d + C(q_d,\dot{q}_d,\theta)\dot{q}_d + g(q_d,\theta) - Q(q_d,\dot{q}_d)] + q_d \tag{18}$$

then control law (7), (8), (18) and (10) ensures that a well–designed flexible joint robots with complete dissipation will track the desired trajectory asymptotically.

The complete dissipative torques $Q(q,\dot{q})$ can be realized by using mechanical dampers – their outputs are the viscous friction torques which can be expressed as:

$$Q(q,\dot{q}) = -\overline{Q}(q,\dot{q})\dot{q} \tag{19}$$

where $\overline{Q}(q,\dot{q})$ is a uniformly positive definite matrix for all q and $\dot{q}$. The linearity of the dampers is not required here.

Similar conclusions can be drawn regarding the control law with mechanical dampers, except that the addition of the dampers to the robots will benefit the robotic system design. It is proved [29] that the recognition of the dissipation in nonlinear systems will greatly help the stability analysis of the control systems.

The advantages of control law (7) – (10) over the existing control laws are that (a) only the measurement of p, $\dot{p}$, q is needed to implement it; (b) p_d, $\dot{p}_d$, $\ddot{p}_d$ can be computed off–line; and (c) the accuracy of the parameters in $M(q,\theta)$, $H(q,\dot{q},\theta)$ only affect the trajectory tracking accuracy of the joint angles, but not the stability of the robotic control systems. Moreover, it is simple and easy to implement the control laws in real–time.

If setting point control is expected, i.e., q_d= *constant*, then a much simpler control law can be derived from (7) – (10):

$$u = -I_m[D_1\dot{p} + D_2(p - K^{-1}g(q_d) - q_d)] + B_m\dot{p} + F_m - K(q-p) \tag{20}$$

It is worth mentioning that in the real world, no absolutely *rigid* robot exists. The so–called *rigid* robots can be modelled as flexible joint robots with large joint stiffness. Therefore, by adjusting some mechanical parameters to make them well–designed, the above control laws can also be applied to control the so–called *rigid* robots, and the same benefits can be obtained.

Actually, we can intentionally introduce high joint flexibility to robots, firstly, to achieve simple control law design. Secondly, joint flexibility becomes a useful feature when the robots contact the environment or manipulate on objects – they function more like artificial arms. Destructive collision can be avoided due to the high joint flexibility, which makes the robots possess some desired impedance. Therefore, only motion or position control design is needed for such robots. When the robots contact the environment, the motion control works as the impedance control.

Modifying the Control Law for Robots with Unknown Parameters

When I_m, B_m, F_m and K are unknown, an adaptive version of the control law (7)–(10) can be easily derived. When the parameters of $M(q,\theta)$, $H(q,\dot{q},\theta)$ are of high uncertainty, the desired trajectory p_d defined by Eq. (9), or (18) can be generated through iterative learning control schemes. These will be addressed in the following.

A. Adaptation of control law (7)–(10) when I_m, B_m, F_m and K are unknown

Define $s = \dot{\delta} + \Lambda\delta$, where all the eigenvalues of Λ are positive. Therefore, $s \to 0$ means $\delta \to 0$. Moreover,

$$s = \dot{p} - \dot{p}_r \quad \text{and} \quad \dot{p}_r = \dot{p}_d - \Lambda\delta \tag{21}$$

$\dot{p}_r$ is called the reference motor velocity. Substituting Eq. (21) into Eq. (2) gives:

$$I_m\dot{s} + B_m s + Y(q,\ p,\ \dot{p}_r,\ \ddot{p}_r)\phi = u \tag{22}$$

$$Y(q,\ p,\ \dot{p}_r,\ \ddot{p}_r)\phi = I_m\ddot{p}_r + B_m\dot{p}_r + F_m - K(q-p) \tag{23}$$

where vector ϕ contains all the unknown parameters in I_m B_m F_m and K; $\hat{\phi}$ stands for its estimation; $Y(q,\ p,\ \dot{p}_r,\ \ddot{p}_r)$ is a known matrix. Define the adaptive control law as:

$$\begin{aligned} u &= Y(q,\ p,\ \dot{p}_r,\ \ddot{p}_r)\hat{\phi} - Ds \qquad (24)\\ &= \hat{I}_m\ddot{p}_r + \hat{B}_m\dot{p}_r + \hat{F}_m - \hat{K}(q-p) - D\dot{\delta} - D\Lambda\delta \end{aligned}$$

where matrix D is symmetric positive definite. It is simple, similar to Eq. (7). The corresponding parameter estimation law is

$$\dot{\hat{\phi}} = -\Gamma Y(q,\ p,\ \dot{p}_r,\ \ddot{p}_r)^T s \tag{25}$$

with matrix Γ symmetric positive definite.

Given the following Lyapunov function candidate:

$$V(t) = \frac{1}{2}[s^T I_m s + \tilde{\phi}^T\Gamma^{-1}\tilde{\phi}] \tag{26}$$

where $\tilde{\phi} = \hat{\phi} - \phi$. It is easy to prove (following [22]) that the adaptive control law (24) and (25) makes $s \to 0$ as $t \to \infty$. The corresponding error equations are:

$$I_m\dot{s} + (B_m + D)s = Y(q,\ p,\ \dot{p}_r,\ \ddot{p}_r)\tilde{\phi} \tag{27}$$

Similarly, eqs. (27) and (12) form BIBO equations, and adjusting D and Λ can make the well–designed flexible joint robots possess desired performance. All the discussions in the previous section apply here.

B. Estimation of p_d through iterative learning

It can be seen from Eq. (9) or (18) that p_d is related to $M(q_d,\theta)$, $H(q_d,\dot{q}_d,\theta)$. When the parameters in $M(q,\theta)$, $H(q,\dot{q},\theta)$ are of high uncertainty, p_d can not be calculated using Eq. (9), or (18). Here, the iterative learning control scheme proposed by Arimoto et al [24–25] is adopted to determine p_d and make $q \to q_d$. Therefore, p_d can be learned iteratively using the following PID learning law:

$$p_{di+1}(t) = p_{di}(t) + Qe_i(t) + R\dot{e}_i(t) + S\int_0^T e_i(\tau)d\tau \quad (28)$$

where the subscript i indicates the quantity in the i–th iteration; $e(t)=q(t) - q_d(t)$; Q, R and S are the constant matrices which determine the learning rate. For better performance of convergence, higher order PID learning laws [26] can be adopted:

$$p_{di+1}(t) = \sum_{k=1}^{N} (P_k p_{di-k+1}(t) + Q_k e_{i-k+1}(t) +$$

$$R_k \dot{e}_{i-k+1}(t) + S_k \int_0^T e_{i-k+1}(\tau)d\tau \,) \quad (29)$$

where N is the order of the iterative learning control algorithm; P_k, Q_k, R_k and S_k are the constant learning parametric matrices of proper dimensions.

Properly choosing the above learning matrices can make $p_{di}(t) \rightarrow p_d(t)$ for $t \in [0, T]$ as $i \rightarrow \infty$. Integrating the above modifications provides highly robust control laws for well–designed flexible joint robots with unknown parameters.

Simulation Results

In order to demonstrate the proposed control algorithms, two robot manipulators with flexible joints are studied. One is a two–link planar manipulator in horizontal plane, the other is a two–link planar manipulator in vertical plane. Their dynamical equations of motion can be expressed as follows:

$$(A_1 + A_2c_2)\ddot{q}_1 + (A_3 + A_2c_2/2)\ddot{q}_2 - A_2\dot{q}_2(\dot{q}_1 + \dot{q}_2/2)s_2$$

$$+ A_4s_1 + + A_5s_{12} + K_1(q_1 - p_1) = 0 \quad (30)$$

$$(A_3 + A_2c_2/2)\ddot{q}_1 + A_3\ddot{q}_2 + A_2\dot{q}_1^2\, s_2 + A_5s_{12}$$

$$+ K_2(q_2 - p_2) = 0 \quad (31)$$

$$I_{m1}\ddot{p}_1 + B_{m1}\dot{p}_1 + F_1\, sat(\dot{p}_1) - K_1(q_1 - p_1) = u_1 \quad (32)$$

$$I_{m2}\ddot{p}_2 + B_{m2}\dot{p}_2 + F_2\, sat(\dot{p}_2) - K_2(q_2 - p_2) = u_2 \quad (33)$$

where $s_i = \sin q_i$, $c_i = \cos q_i$ and $sat(.)$ is the saturation function.

For the robot in horizontal plane,

$A_1 = 6.42\ kgm^2 \qquad A_2 = 2.10\ kgm^2$

$A_3 = 0.85\ kgm^2 \qquad A_4 = A_5 = 0$

$I_{m1} = 1.94\ kgms^2 \quad B_{m1} = 1.10\ Nms/rad$

$I_{m2} = 0.25\ kgms^2 \quad B_{m2} = 0.22\ Nms/rad$

$F_1 = 0.5\ Nm \qquad K_1 = 790\ Nm/rad$

$F_2 = 0.5\ Nm \qquad K_2 = 610\ Nm/rad$

For the robot in vertical plane,

$A_1 = I_{L1} + I_{L2} + m_pL_2^2 + L_1^2(m_{L2} + m_j + m_p)$

$A_2 = 2L_1L_2(m_{L2}/2 + m_p)$

$A_3 = I_{L2} + m_pL_2^2$

$A_4 = gL_1(m_{L1}/2 + m_{L2} + m_j + m_p)$

$A_5 = gL_2(m_{L2}/2 + m_p)$

where I_{Li} is the moment of inertia of link i, I_{mi} is the motor moment of inertia of the joint i, m_{Li} is the mass of link i, L_i is the length of link i, m_j is the mass of joint 2, m_p is the mass of payload, g is the gravitational acceleration. A set of nominal values are

$I_{m1} = I_{m2} = 0.2\ kgms^2/rad$

$B_{m1} = B_{m2} = 0.00573\ Nms/rad$

$F_1 = F_2 = 0.5\ Nm \quad K_1 = 5\ Nm/rad$

$K_2 = 4.5Nm/rad \quad m_{L1} = m_{L2} = 3kg$

$m_j = 0.1kg \quad m_p = 1kg \quad L_1 = L_2 = 1m$

Since the horizontal–plane robot is *well–designed*, the setting point control law (20) is applied to it with T=0.01 s, $D_1 = D_2 = 4I$ (I is identity matrix). Let q_{d1} =1.5 *rad*, q_{d2}=–0.5 *rad*. The time responses under nominal parameters are shown in Fig. 5. Even when K_1 becomes 600, K_2 becomes 510, A_1 becomes 4.42, A_2 becomes 1.85, and A_3 becomes 1.05, and the controller remains same without adaptation and learning, the control result is almost the same as shown in Fig. 5. This proves the high robustness of the control law (20).

The vertical–plane robot is not mechanically *well–designed* from the control point of view. If we apply the setting point control law (20) to it with the same conditions as used in Fig. 5, fluctuating time responses of q_1 and q_2 are obtained. That is because the joints are very soft and the response speed requirement of $D_1 = D_2 = 4I$ is not proper. In order to hold the links at the position of q_{d1} =1.5 *rad*, and q_{d2}=–0.5 *rad*, the difference between p_1 and q_1 should be 15 *rad* (more than two cycles). When $D_1 = 0.5I$, $D_2 = 0.25I$, the setting point control works well, but the response is slow. When $D_1 = 2I$, $D_2 = I$, and K_1 and K_2 are changed to 100 *Nm/rad* and 90 *Nm/rad*, the robot becomes well–designed. The corresponding time responses are similar to Fig. 5. As stiffness K_1 and K_2 increases, the responses becomes better. Again, the control system is not sensitive to the parameter errors. This simulation also suggests that flexible joint robots with variable structures can improve the control performance by adjusting its structural parameters.

The following simulations are performed to verify the trajectory tracking control law (7) – (10) and the vertical–plane robot was used. To show that there are many freedoms to modify robotic structural parameters, the mechanical parameters of the vertical–plane robot are changed to $m_{L1} = m_{L2} = m_p = 0.5kg$, $L_1 = L_2$= 0.5*m*, with all other parameters being nominal values. Then the control law (7) – (10) is applied to achieve the desired trajectory defined by $q_{d1} = sin\ (\pi t/5)$, and $q_{d2} = 1 - q_{d1}$. Very good results were obtained as shown in Fig. 6. When the initial trajectory errors are set to 1 *rad*, we still got good results. This means that the constraints on initial trajectory errors are not strong. If only K_1 and K_2 are changed to 100 *Nm/rad* and 90 *Nm/rad* and all other parameters assume their nominal values, we can also obtain good results similar to Fig. 6. As K_1 and K_2 increases, the responses becomes better. Therefore, it is possible to find an optimal design of the robot using the proposed integrated design procedure.

When we add complete dissipative torques $Q(q, \dot{q}) = -10\dot{q}$ to the robot, very good results were obtained with all the parameters assuming their nominal values. This means that adding mechanical dampers can also make a robot well–designed.

These simulations show that changing mechanical structural parameters can make a robot well–designed. The new control laws are not sensitive to dynamical friction torques in (32) and (33). Because of the simplicity of this approach, the corresponding control laws are not sensitive to the sampling frequency. Good results can still be obtained even when the sampling period T was raised to $0.05s$. For the vertical–plane robot, increasing K_1 and K_2, or decreasing m_{L1}, m_{L2}, m_p, L_1, *or* L_2 (mechanical parameters) and adjusting D_1, D_2 (control parameters) can always ensure good tracking performance. Concurrently searching the best mechanical and controller parameters will greatly improve system controllability.

Conclusion

In this paper, the basic idea of integrated mechanical and control design for well–designed flexible joint robots is proposed, which appears to be very useful and will be further explored. The concept of variable structure robots provide the possibility to make a robot well–designed for different tasks. Compared with the existing control schemes for flexible joint robots, the simple partial inverse control laws do not need the measurement of accelerations and jerks and are suitable for real–time implementation. These control laws are also robust to the robotic model errors. This approach allows further adaptive control design based on the robotic structure parameters θ. We hope that this paper encourages more work on this integrated mechanical and control design approach for robots.

References

[1] A. Ficola, R. Marino, and S. Nicosia, "A singular perturbation approach to the control of elastic joints", Proc. 21st Ann. Allerton Conf. on Communications, Control and Computing, Monticello, Ill., 1983, pp.220–225.

[2] B. Siciliano, and W.I. Book, "A perturbation approach to control of lightweight flexible manipulators", Int. J. Robotics Res., 7(4):79 – 90, 1988.

[3] M.W. Spong, J.Y. Hung, S.A. Bortoff, and F. Ghorbel, "A comparison of feedback linearization and singular perturbation techniques for the control of flexible joint robots", Proc. American Control Conf., Pittsburgh, PA, 1989, pp.25–30.

[4] M.C. Readman and P.R. Belanger, "Analysis and control of a flexible joint robot", Proc. 29th IEEE CDC, Hawaii, 1990.

[5] F. Ghorbel, J.Y. Hung, and M.W. Spong, "Adaptive control of flexible–joint manipulators", IEEE Contr. Sys. Magazine, Dec. 1989, pp.9–13.

[6] K.–P. Chen and L.–C. Fu, "Nonlinear adaptive motion control for a manipulator with flexible joints", Proc. 1989 IEEE Int. Conf. Robotics & Automat., Scottsdale, Arizona, May 1989, pp.1201–1206.

[7] J. Yuh and D.K. Tissue, "Adaptive control for mechanical manipulators having a joint compliance", Proc. 1990 IEEE Int. Conf. on Robotics and Automation, Cincinnati, OH, pp. 1632–1637.

[8] K. Khorasani, "Adaptive control of flexible joint robots", Proc. 1991 IEEE Int. Conf. on Robotics and Automation, Sacramento, CA, Apr. 1991, pp.2127–2134

[9] K. Khorasani and M.W. Spong, "Invariant manifolds and their application to robot manipulators with flexible joints", Proc. 1985 IEEE Int. Conf. Robotics & Automation, St. Louis, MO., 1985.

[10] M.W. Spong, "Modelling and control of elastic joint robots", Trans. ASME, J. Dynamic Sys., Meas., Control, 109:0–9, 1987.

[11] M. W. Spong, "On the force control problem for flexible joint manipulators", IEEE Trans. on Automatic Control, vol. 34, no. 1, Jan. 1989, pp.107 – 111.

[12] S. Nicosia and P. Tomei, "On the feedback linearization of robots with flexible joints", Proc. 27th CDC, Austin, Texas, Dec. 1988, pp.180–185.

[13] K.P. Jankowski and H.A. ElMaraghy, "Dynamic decoupling for hybrid control of rigid/ flexible joint robots interacting with the environment", Proc. 1991 IEEE Int. Conf. on Robotics and Automation, Sacramento, CA, Apr. 1991, pp. 1226 – 1231.

[14] S.A. Bortoff and M.W. Spong, "Feedback linearization of flexible joint manipulators", Proc. 1987 IEEE Conf. on Decision & Control, pp.1357 – 1362.

[15] C. Canudas De Wit and O. Lys, "Robust control and parameter estimation of robots with flexible joints", Proc. 1988 IEEE Int. Conf. on Robotics & Automat., pp.324–329.

[16] L.–C. Lin, "State feedback H_∞ control of manipulators with flexible joints and links", Proc. 1991 IEEE Int. Conf. on Robotics & Automat., Sacramento, CA, Apr. 1991, pp.218 –223.

[17] J.–J.E. Slotine and S. Hong, "Two–time scale sliding control of manipulators with flexible joints", American Contr. Conf., 1986, pp. 180–186.

[18] F. Liang and H.A. ElMaraghy, Robust control of flexible joint robots, to appear in Proc. 1993 Canadian Conf. on Electrical and Computer Engineering, Vancouver, Sept. 1993.

[19] F. Miyazaki, S. Kawamura, M. Matsumori, and S. Arimoto, "Learning control scheme for a class of robot systems with elasticity", Proc. 25th CDC, Athens, Greece, 1986, pp.74–79.

[20] M. W. Spong and M. Vidyasagar, Robot Dynamics and Control, John Wiley & Son, 1989.

[21] N. P. Suh, The Principles of Design, Oxford University Press, 1990.

[22] J.–J.E. Slotine and W. Li, Applied Nonlinear Control Systems, Prence–Hall, 1991.

[23] N. Rouche, P. Habets and M. Laloy, Stability Theory by Liapunov Method, Springer–Verlag, 1977.

[24] S. Arimoto, S. Kawamura, and F. Miyazaki, Bettering operation of robots by learning, J. of Robotic Systems, Vol. 1, Summer, 1984, pp. 123–140.

[25] S. Arimoto, F. Miyazaki, and S. Kawamura, Motion control of robotic manipulators based on motor program learning, 2nd IFAC Symp. on Robot Control, 1988, pp. 169–176.

[26] Z. Bien and K.M. Huh, Higher-order iterative learning control algorithm, IEE Proc.–D, 136(3), 1989, pp. 105–112.

[27] J. Imura, T. Sugie, and T. Yoshikawa, Global robust stabilization of nonlinear cascaded systems, IEEE 31th Conf. on Decision and Control, Tucson, Arizona, 1992, pp. 3031–3036.

[28] Z. Lin and A. Saberi, Semi-global stabilization of partially linear composite systems via feedback of the state of the linear part, IEEE 31th Conf. on Decision and Control, Tucson, Arizona, 1992, pp. 3431–3433.

[29] D.J. Hill, Dissipative nonlinear systems: basic properties and stability analysis, IEEE 31th Conf. on Decision and Control, Tucson, Arizona, 1992, pp. 3259–3264.

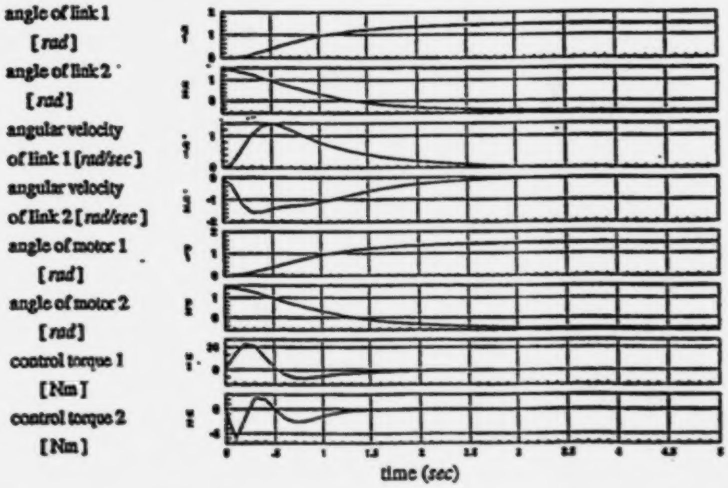


Fig. 5 Time responses of the horizontal-plane two-link robot under control law (20)

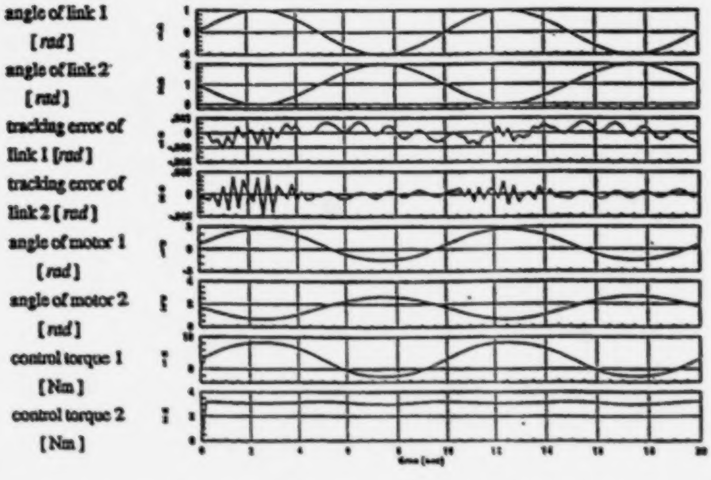


Fig. 6 Time responses of the vertical-plane two-link robot under control law (7) – (10)

y0324C

Remote Mine-Hunting Systems - Vehicle Stability Trials

Jon M. Preston
Mine Countermeasures Group,
Defence Research Establishment Pacific,
CFB Esquimalt, Victoria, BC V0S 1B0

Landy Shupe
International Submarine Engineering Research Ltd.
1734 Broadway St.
Port Coquitlam, BC V3C 2M8

#149633

Abstract

The Maritime Coastal Defence Vessels (MCDV) will allow the Canadian Navy to perform detailed route survey by acquiring, mosaicing, and cataloguing high resolution sonar images of the coastal sea bottom. The MCDV sonar system will be a side-scan sonar operating from a towfish. Mine hunting, which would be based on comparisons between new and catalogued images, could be done with similar equipment and procedures, except that the MCDV could not safely enter a minefield. A remote system for mine hunting could potentially be faster and more flexible, as well as safer.

An important issue in the design of remote systems for acquiring sonar imagery is that the sonar towfish must be stable, that is, must not respond excessively to the wave-driven motion of the towing vessel. Even modest motions of a sonar platform can distort images and generate coverage gaps. Comparative measurements of the wave-driven motions of a surface drone, of a near-surface drone, and of a ship representative of an MCDV were performed off the west coast of Vancouver Island. The results show that a near-surface drone is as stable as a ship comparable to an MCDV except in surge. The effects of DOLPHIN surge could probably be significantly reduced by driving the winch in opposition to the surge velocity, but this has not been demonstrated.

Background

Mine hunting is the process of finding mines, normally on or close to the sea bottom. An important mine countermeasure, it provides intriguing technical challenges. There are four steps: detection of mine-like objects, classification that they are likely mines, identification to prove that they are mines, and disposal. The first two require sonar, the last two a diver or vehicle. Together with mechanical, electromagnetic, and/or acoustic mine sweeping, mine fields can be cleared to a high level of confidence.

The traditional approach to mine hunting employs a specialised ship, such as the Royal Navy's Single Role Minehunter [1]. Key components of these sophisticated vessels include a complex hull-mounted (or variable- depth) sonar, command and control systems for holding a position or moving the ship in any direction or orientation, and underwater vehicles for identification and disposal. To enable the ships to operate close to live mines, they are designed both to withstand large shocks and for minimal magnetic and acoustic signatures, which implies strongly braced non-metallic hulls, specialised alloys, and careful attention to equipment mounts. The mine hunting procedure is to search ahead of the ship by sonar to detect mine-like objects, then to switch the sonar to higher resolution for classification (often manoeuvring the ship to a different perspective), then to launch the identification and disposal vehicle if appropriate. These ships and procedures have been used and refined for decades, and are of proven effectiveness.

The vessels being built for the Canadian Navy for mine countermeasures, the Maritime Coastal Defence Vessels (MCDV) will also have other missions, such as coastal patrol. They are not designed to enter a minefield safely. Because of this, and for other reasons discussed below, remote mine hunting systems are being considered.

Remote Mine Hunting

Remotely controlled or autonomous mine hunting systems are being considered even by navies

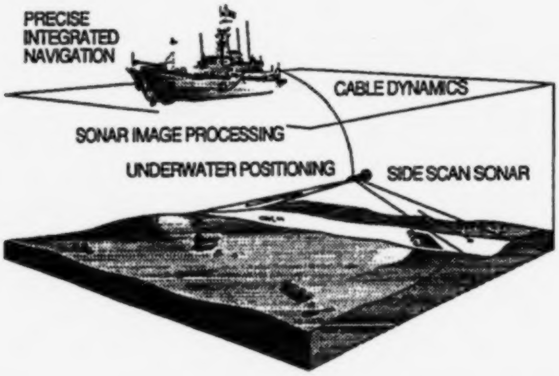


Figure 1: *Equipment and concepts used for route survey at DREP.*

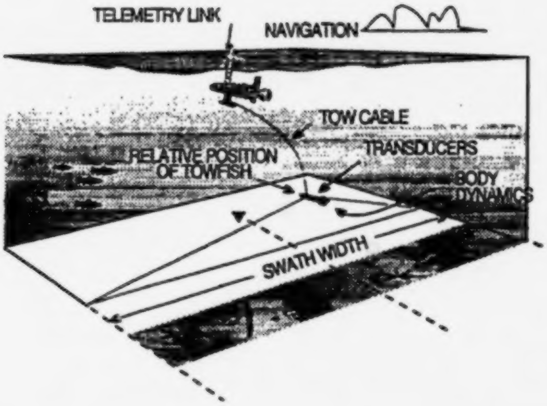


Figure 2: *A concept for minehunting with a drone and a towfish.*

with large commitments to mine hunting ships. The obvious benefit is increased safety. The requirements and costs for signature reduction, blast hardening, and manouverability are far less for a ship which is not designed to enter a minefield. Because unmanned systems could safely pass close to the mine there is another important benefit: the sonar need not be as sophisticated to operate at ranges of about 100 m as it would for about 600 m.

A promising concept for remote mine hunting is similar to the procedures used for route survey. Route survey is the peacetime process of acquiring sonar images of the entire area in which mine hunting is anticipated, to provide a reference against which mine-hunting images can be compared. Figure 1 illustrates acquiring such images with a sidescan sonar on a towfish. The sonar range is typically 100 m, and the towfish altitude is ideally about 15% of the range, so that objects generate clear shadows. Route survey will be a common task on an MCDV [2], and it will be done as shown in Figure 1.

One concept for remote mine hunting is shown in Figure 2, obviously quite similar to Figure 1. The ship has been replaced by a drone, and a high-bandwidth link transmits the sonar images to the command vessel or shore-based command centre. The other concepts for remote minehunting involve powered submerged sonar-carrying vehicles, either swimming ahead of the ship or untethered and with varying degrees of autonomy. All remote mine hunting systems which have progressed beyond the conceptual stage would do only the sonar stages of mine hunting: detection and classification. Unlike ship-based systems, they would continue on their search tracks rather than stop to identify or dispose of a mine-like object. Knowing the coordinates of the classified objects, the MCDV commander would then plan a safe approach to each object for identification and disposal with a remotely operated vehicle.

Several challenges arise in the design of a drone-towed remote mine-hunting system, particularly navigation, the telemetry link, and maintaining towfish altitude and stability. This paper deals with towfish stability, that is, the design of a towfish and tow system to reduce the wave-driven motion of the towfish to acceptable levels.

Stability of Sonar Towfish

The wave-driven motion of the towing ship or drone readily causes towfish motion. Some towfish motions, particularly yaw, markedly degrade the sonar image and can make detection and confirmation impossible [3], [4]. Among the techniques to mitigate this problem are active control surfaces on the towfish [5] (and several commercial towfish), towfish design [6], and different tow configurations such as the two-part tow [3]. This is a particularly relevant question for remote mine hunting systems since the drones will be far smaller than ships, and have very different wave-driven motions.

With a towed system there are only two choices for drones: surface vessels and near-surface vessels. They would be expected to have quite different motions in waves. The only near-surface drone we know of is DOLPHIN (Deep Ocean Logging Platform for Hydrographic Instrumentation and Navigation), developed, manufactured, and operated by International Submarine Engineering Research Ltd, Port Coquitlam, BC. DOLPHIN, shown in Figure 3, is a diesel-powered snorkelling submersible, 7.6 m overall and 3.3 t, which normally travels 3 m below the surface, inhaling combustion air through the snorkel. It was originally developed for hydrographic survey, as a vessel which would be far more stable than a surface launch, because it travels beneath the waves and has active stability control. DOLPHIN's control system is a two-layer hierarchical controller operating at 50 Hz [7]. A small fleet of DOLPHINs is currently performing hydrographic survey off Newfoundland.

Most of the engineering challenges associated with a remote mine-hunting system are the same for surface and near-surface drones. Examples are navigation [7], telemetry, and the sonar system. The most important difference is stability in rough seas. To justify the additional complexity of a near-surface system it would have to be shown that their stability is considerably better.

Measuring Wave-Driven Motion

A trial was conducted to measure the motions of candidate surface and near-surface drones, and a ship. Because of the variability of waves, meaningful comparisons can be made only if all vessels operate simultaneously in the same waves. The ship was the *Vector*, 500 t and 40 m in length, normally used for hydrographic research, and somewhat smaller than an MCDV. Drone vessels have been developed by Germany and Denmark. The German Troika remote-control vessel is 25 m overall and 96 t [8], and the Danish Stanflex system [9] uses vessels 18 m long displacing 32 t. A long-standing concept in MCM is to use craft of opportunity, that is, to requisition available craft. A sensible choice for mine hunting in exposed waters is a commercial fishing vessel, most of which have hull designs which emphasize sea-keeping. To represent a typical trawler, and also to give an indication of the wave-driven motions of the Troika and Stanflex drones, we used a fishing vessel named *Caligus*, 65 t and 15.5 m. Even *Caligus* was unable to put to sea safely on the roughest day of the trial, and a smaller vessel would likely have been even more restricted.

Each of the three vessels was instrumented to measure motion of the tow point, as indicated in Figure 3 for DOLPHIN. As well as its regular suite used for vehicle control, DOLPHIN carried a gyroscopically stabilized package (Humphrey Inc. SA09-0601-1) which measured the three acceleration components plus roll and pitch, to provide improved data. The remotely controlled keel-mounted winch included sensors for revolutions paid out and cable tension. When a depressor was used, either deadweight or hydrodynamic, its depth was measured with a pressure sensor. The DOLPHIN-towed towfish, which was 22 cm in diameter and 1.7 m long, with a total fin area of 0.3 m^2, carried a full instrumentation suite including another Humphrey SA09-0601-1, plus heading, pressure, angular rate, and strapdown acceleration sensors.

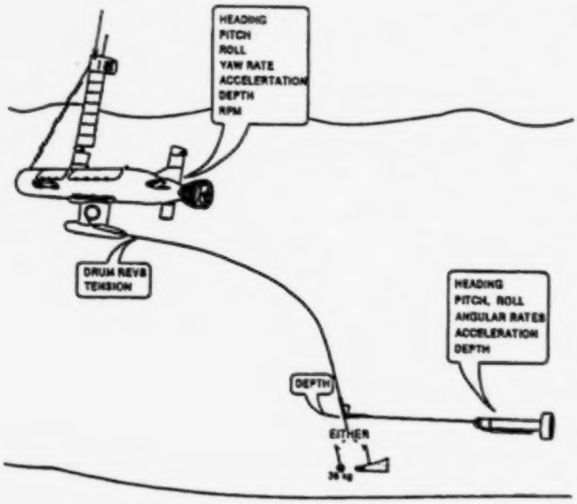


Figure 3: *The DOLPHIN vehicle and its towfish, towed as a two-part tow. On some trials the towfish was ballasted and towed directly, on others either a 36-kg lead ball or a hydrodynamic depressor was used. Sensor locations are indicated.*

Each ship carried, near the tow point, a Ship Motion Sensor System (SMSS), developed and operated by Defence Research Establishment Atlantic (DREA). SMSS is based on a similar triaxial gyro-stabilized accelerometer, plus angular rate sensors. Our only towfish instrumentation suite was on the DOLPHIN-towed towfish, so motions of the ship-towed towfish were sensed only with pressure sensors on their depressors. (Because a depressor and two-part tow significantly increases the stability of ship-towed towfish [10], two-part tows were used throughout the trial with the ships.)

Each day all three vessels left harbour for the trials area, where they followed a race track course as shown in Figure 4. Of the eighteen pre-programmed courses, the one was selected which was best oriented into the swell, to provide the five wave aspects shown in Figure 5. The waverider

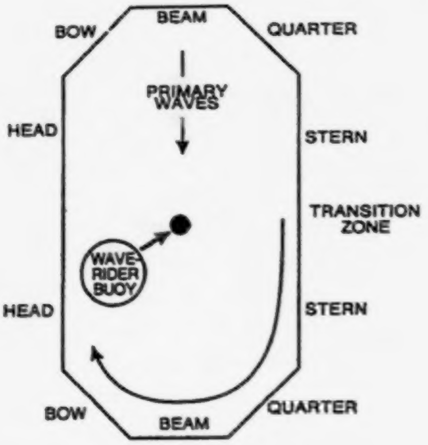


Figure 4: *A race track pattern sailed clockwise by the vessels to generate motion data for five wave aspects. Each circuit produced two sets of aspects.*

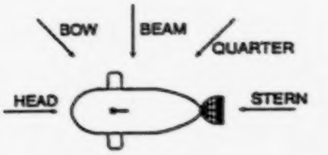


Figure 5: *Definitions of the five wave aspects.*

buoy deployed for the trial reported wave heights and periods; swell was from the west each day with very little wind-driven waves, thus the wave spectrum generally had but one peak. Cable lengths and ship speeds were changed within the transition zone, but the tow configuration could not be changed at sea.

Data Analysis

Each sensor (pressure, pitch, ...) produced a time series for each wave aspect under each condition. The averaging process used to characterise them was the so-called significant average. This average is well established in wave statistics [11], and was calculated each half-hour by the package supplied with the wave-height measuring buoy, the Waverider. It is the mean of the largest one-third of the vertical distances between trough and crest, designated $\overline{H_{1/3}}$. The same mean was calculated for each time series (heave, roll, etc.).

Displacements are more intuitive than accelerations, and can be compared directly with wave heights, therefore the vertical, fore/aft, and lateral accelerations were integrated twice to give heave, surge, and sway respectively. Pitch rate was calculated by numerical differentiation of pitch. Filtering to remove offsets, drifts, and artifacts was a major part of the data processing and required considerable refinement to generate results which were consistent and merited confidence.

The head aspect was selected as the aspect at which the primary inter-vessel comparisons were made. Data from the other aspects have been plotted with respect to the same motion in head seas. The amount of data taken for each speed and wave aspect was about 12 min, in which time an average of 60 waves were encountered (less data were recorded in sea state 6).

Results

Head Seas

It was found that the significant means of the vessel and drone displacements in head seas were nearly proportional to the significant wave height. The same proportionality was found for the products of the significant means of pitch, pitch rate, and roll multiplied by the mean wave period. Figure 6 is a plot against $\overline{H_{1/3}}$ of the significant mean surge divided by $\overline{H_{1/3}}$ and Figure 7 is similar for the product of significant-mean pitch and mean wave period. Over this range of sea states those ratios do not appear to depend on wave height or speed. The vessels surge with very different amplitudes with, surprisingly, *Caligus* far less than *Vector*.

Mean values for all motions are listed in Table I. DOLPHIN's roll, pitch, and sway are notably more stable than either ship, as might be expected for a vehicle with active control [7]. Heave is not as well controlled because DOLPHIN maintains depth rather than minimises vertical motion, and this distinction is significant in long-wavelength swells. DOLPHIN does not control its surge.

Other Wave Aspects

The aspect dependence of the variables listed in Table I have been calculated and two examples, surge and roll, are shown in Figures 8 and 9. Each data point was calculated by dividing the significant mean of that quantity at that aspect and speed by the significant mean of the same quantity in head seas at the same speed. The curves in these plots are spline curves fitted to mean values at each aspect, and have no theoretical significance. Some comments on the corresponding plots for the other

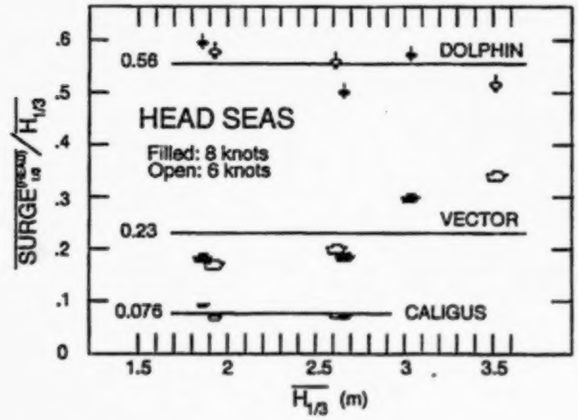


Figure 6: *The ratio of significant-mean surge of the vessels and drone to significant wave height plotted against significant wave height. Vector data are plotted as ship silhouettes, Caligus data as boats, and DOLPHIN data as its distinctive shape.*

Table 1: SHIP MOTIONS IN HEAD SEAS

	DOLPHIN	*Vector*	*Caligus*
Heave	0.78	1.07	0.94
Surge	0.56	0.23	0.076
Sway	0.39	0.56	0.71
Pitch	12.7	25.7	34.4
Roll	10.8	19.2	17.3
Pitch Rate	12.4	32.2	64.4
Yaw Rate		3.6	8.6

Values shown are the average over all trials of the significant mean of that quantity divided by the significant wave height; for angles (deg s/m) and angular rates (deg/m) the significant mean was also multiplied by the mean wave period at the time of the measurement. Yaw rate was not measured on DOLPHIN.

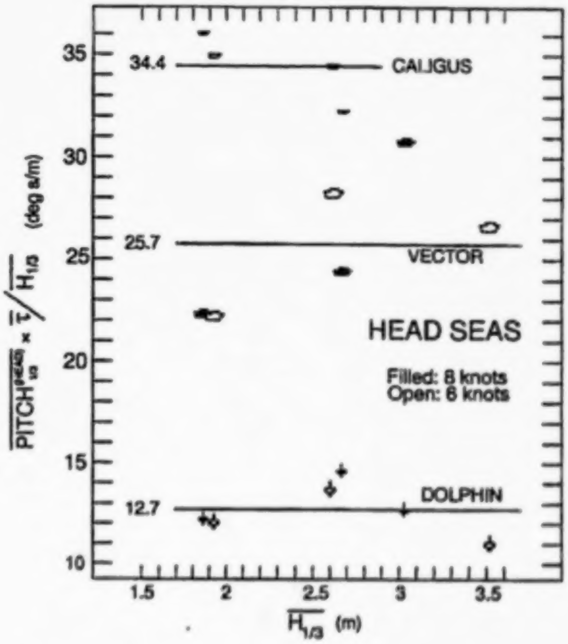


Figure 7: *The ratio of significant-mean pitch of the vessels and drone multiplied by the wave period during that measurement to significant wave height plotted against significant wave height. Symbols as in previous figure.*

motions, listed in Table I, follow, even though their plots are not reproduced here. Heave had the least aspect dependence, for all vessels. Roll and sway were approximately equal for head and stern aspects, and considerably larger at or near beam, particularly for the ships. Surge and pitch showed clear aspect variations between vessels: *Vector* surged most in head seas while the lighter surface vessel, *Caligus*, and DOLPHIN surged far more in stern seas. The ships had maximum pitch in head seas while DOLPHIN pitched most in stern seas, but even in stern seas the ships pitched almost as much as DOLPHIN: *Vector* 67% and *Caligus* 87%.

In these ship motion and attitude results, DOLPHIN is found to be at least as stable as the ships except in surge, and far more stable for some variables. However, as shown in examples such as Figure 10, the DOLPHIN-towed towfish clearly undergoes more vertical motion than those towed by the ships (the three tow configurations were identical, using a downweight and neutrally buoyant towfish).

Consideration of the cable dynamics shows that surge is, for DOLPHIN's cable configuration in this trial, the drone motion which most affects cable tension. Chapman [12] supports the sheath model as an accurate description of cable response to wave-driven motion. In this model, motions of the top of the cable are divided into two components, parallel and transverse to the upper part of the cable. Parallel motions are transmitted down the cable without (in the short term) changing the cable profile, and transverse motions excite waves which become insignificant after only a few tens of metres, thus the cable moves as if it were in a sheath. Furthermore, the cable leaves DOLPHIN

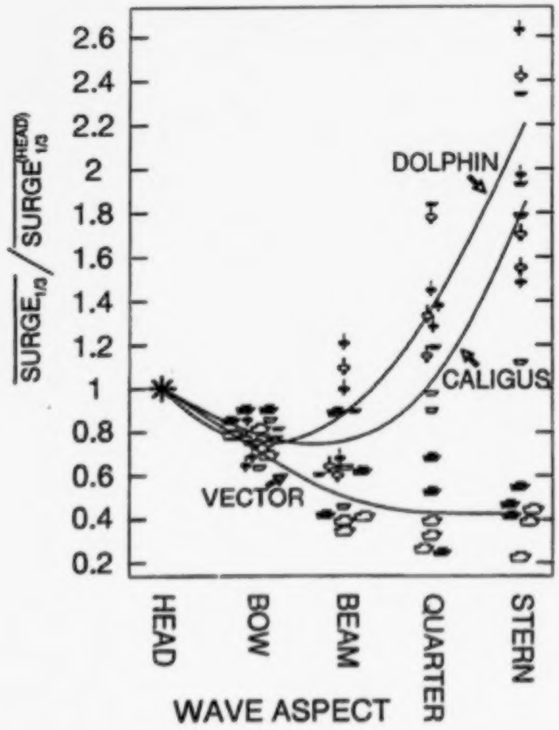


Figure 8: *Significant-mean surge of the vessels and drone at various wave aspects compared to surge in head seas. Curves are spline fits with no theoretical significance. Symbols as in previous figures.*

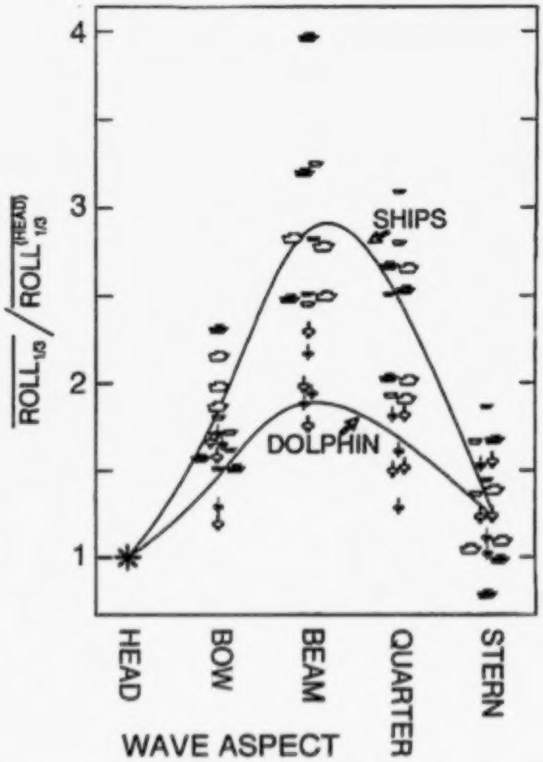


Figure 9: *Significant-mean roll of the vessels and drone at various wave aspects compared to roll in head seas. Curves are spline fits with no theoretical significance. Symbols as in previous figures.*

at no more than 15° below the horizontal, but is close to vertical at the depressor. Thus DOLPHIN surge efficiently drives vertical towfish motion. Figure 11 clearly shows the high correlation between DOLPHIN forward velocity and downweight vertical velocity. Velocities were plotted, rather than motions or accelerations, because cable tension is proportional to velocity squared, and the tension, measured where the cable exited DOLPHIN, is indeed tightly correlated with these velocities.

Ships do not surge nearly as much (Figures 6 and 8), and their tow cables are further from the horizontal because the uppermost parts are in air, making surge not the only important motion.

Thus DOLPHIN was found to be at least as stable as the ships in the motions and attitudes which drive the motions of ship-towed towfish. However it was shown that surge is particularly important for a DOLPHIN-towed towfish, and that DOLPHIN surges considerably more than either ship.

It should be possible to compensate for DOLPHIN surge by sensing DOLPHIN fore-aft acceleration and driving the winch to cancel DOLPHIN's surge velocity. The existing winch is capable of the required speeds.

Conclusions

Wave-driven motions of a towed body can be reduced by compensation for ship motion, by tow configurations such as the two-part tow, by active controls, and by selection of the towing vessel or drone. None of these can eliminate wave-driven motion, so criteria for tolerable motion must be considered. Many factors set these criteria, such as the beamwidth of the sonar, its speed of advance, the target dimensions, and even the intended use of the sonar images. Approximate criteria can be calculated easily but are not generally published, even if they are used for design of towbodies. Reference [4] contains the results of a detailed calculation.

An important decision in the design of a drone-towed remote mine-hunting system is the choice be-

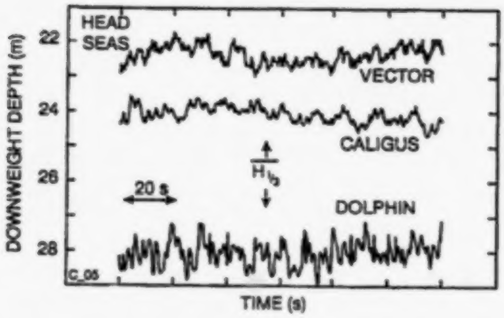


Figure 10: *Depths of the downweights of each two-part tow in head seas. The significant wave height, 1.9 m, is indicated by the labelled vertical arrow.*

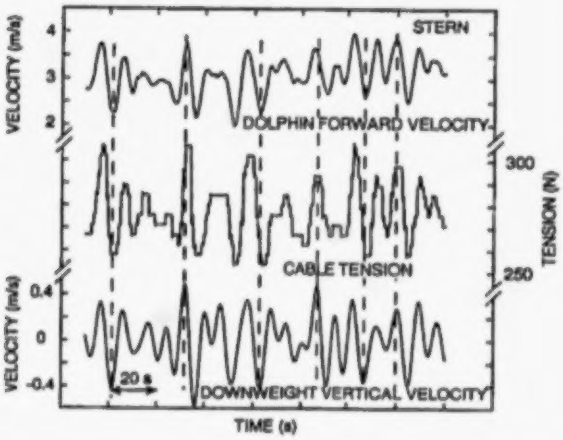


Figure 11: *DOLPHIN speed, cable tension, and downweight vertical speed in stern seas. DOLPHIN speed was calculated by integrating fore/aft acceleration and adding the mean speed. To calculate a credible time derivative of the downweight depth, it was necessary to reduce noise by low-pass filtering below 0.15 Hz. Vertical dashed lines through some extrema of downweight velocity have been added to show correlations.*

tween a surface or near-surface drone. Stability of the towfish against wave-driven motion is a key factor in that decision. Comparative measurements showed that a near-surface drone, DOLPHIN, was at least as stable as much larger surface vessels except in surge. However DOLPHIN's towfish was less stable than the ship-towed fish because the cable dynamics are such that, for a near-surface drone, surge is the wave-driven motion which most effectively drives towfish motion.

While DOLPHIN surge could not readily be controlled, it is likely that the effects of surge could be significantly reduced by driving the winch in opposition to the surge velocity. The winch is capable of the required speeds, but a trial to measure such a reduction has not been done.

REFERENCES

[1] R. Brittle, "The combat system of H.M.S. Sandown," in *Warship '89, International Symposium on Mine Warfare Vessels and Systems 2*, Royal Institute of Naval Architects, 1989.

[2] A. Horowitz, G. Graham, and R. Kuwahara, "The Canadian Maritime Coastal Defence Vessel (MCDV) and the Integrated Survey and Inspection System," in *Oceans '93*, vol. II, pp. 458–462, 1993.

[3] J. Preston, "Stability of towfish used as sonar platforms," in *Oceans '92*, pp. 888–893, 1992.

[4] J. Preston and R. Poeckert, "Distortion and break-up of sidescan images - criteria and reconstruction by geocoding," in *Oceans '93*, vol. I, pp. 371–377, 1993.

[5] J. F. Henderson and R. Wright, "The use of active controls on underwater towed bodies," in *Second Intl. Conf. on Computer Modelling in Ocean Eng.*, 1991.

[6] R. Hubbard, "Hydrodynamic design of a sonar towfish for minimal image-smearing in heavy seas," in *Oceans '89*, 1989.

[7] B. Butler, "Positioning and navigation for the DOLPHIN vehicle using LORAN-C," in *ROV '90*, 1990.

[8] Anonymous, "The Troika simulation sweeping system," *Maritime Defence*, vol. 17, no. 5, 1992.

[9] H. Harboe-Hansen, "Stanflex 300 MCM system update," *Maritime Defence*, vol. 18, no. 5, 1993.

[10] D. Hopkin, J. Preston, and S. Latchman, "Effectivness of a two-part tow for decoupling ship motions," in *Oceans '93*, vol. I, pp. 359–364, 1993.

[11] B. Kinsman, *Wind Waves*. Prentice-Hall, New Jersey, 1965.

[12] D. Chapman, "Effects of ship motion on a neutrally stable towed fish," *Ocean Engineering*, vol. 9, no. 3, 1982.

90C222

Path Planning and Supervised Autonomous Control of a Robotic Arm in Space

H.S. Gaskill & C.P. Dawson†

Department of Mathematics and Statistics, Memorial University, St. John's, NF
INSTRUMAR, Ltd., P.O. Box 13246, Station A, St. John's, NF

#149C34

Abstract

Under Atlantic STEAR initiatives, INSTRUMAR developed a simulation of a planar, three degrees of freedom robotic arm coupled to a path planning system. The planner makes essential use of local experts to define arm motions between via configurations. These concepts and their use in the planner are described. A discussion of their potential use in developing a ground-based control system for a space based robotic arm is given which focuses on problems associated with time lags and error recovery.

1. Introduction

The possibility that the MSS and SSRMS may be controlled from the ground and the types of technologies required to do so has been discussed in the recent paper of Jackson [1992]. Particular technologies which are expected to be required are (see Jackson [1992]):

1. automated work task decomposition;
2. geometrical modelling;
3. trajectory planning;
4. real-time obstacle avoidance;
5. trajectory and force control of redundant manipulators.

An additional priority identified by Immega [1992] is that the operator interface be designed so as to reduce the need to have individual joints of a space robot be manually controlled in real-time.

In Section 2, a methodology for path planning that was developed for the Canadian Space Agency as part of an ACSI STEAR project will be described. The methodology is based on via configurations connected by individual manipulator motion types that are governed by particular local experts. Although this system was designed for a planar 3dof arm, it satisfies many of the requirements identified by Jackson and Immega for space robots, and as such, has application in developing advanced control systems to support teleoperation of the SSRMS.

Section 3 presents a path planning example which illustrates the nature of the plans generated by the path planning system. This example is analyzed to illustrate some of the difficulties encountered when operating a redundant manipulator in manual mode, whether kinematic or dynamic.

Section 4 discusses the utilization of the path planning concepts implicit in the INSTRUMAR algorithm in the development of the technologies identified by Jackson and Immega.

2. Simulator and Path Planner

Brief descriptions of the two main pieces of software developed by INSTRUMAR are given in this section.

The Generic Testing System

The main function of the Generic Testing System (GTS) is to simulate the motion, kinematic or dynamic, of a planar, three degrees of freedom

---

† The authors are grateful to I. Atkinson, H. Spearns and T. Spurrell for their help with software development and to P. Gosse, W. Heath and B. Shawyer for their help with manuscript preparation.

(2D-3dof) robotic arm. Fig. 1 contains a schematic diagram of such an arm in which the two inner links are made flexible by the addition of artificial joints at the mid-points.

Because the simulator was constructed as an aid to evaluating path planning methodologies, it also contains a user interface permitting arm control, and a data base of path planning problems. Fig. 2 shows such a sample path planning problem scenario consisting of an initial arm configuration, a series of polygonal obstacles, and a target position for the end-tip indicated by an ×. By accessing one of the arm motion control menus (not shown), the user can manually attempt to solve the problem. In performing this activity, a collision checking module continually verifies that changes in arm position will not result in an arm configuration which intersects one of the polygonal obstacles. Such configurations are forbidden and attempts to move the arm into such configurations are denied.

The simulator also contains a capability to create sensors of various types, install these sensors on the arm, and to add new problem scenarios to the data base of path planning problems.

The Path Planning System

The other main module, which operates entirely separately from the GTS, is a path planning system (PPS). Because this paper focuses on the application of concepts from the PPS algorithm to the semi-autonomous operation of a space arm using a ground-based control system, we describe the PPS in detail.

Algorithm features

The path planning algorithm employs the following conceptual elements:

- C-space search techniques;
- the use of a penalty function to prioritize target configurations;
- heuristics to limit C-space search volumes;
- the use of local experts to implement human strategies;
- the use of *via configurations*;
- hierarchical organization of planning;
- backwards and forwards chaining.

Details of the use of these concepts are given below.

Initial C-space search

C-space consists of all possible configurations of the robot manipulator under study. In the abstract it would be infinite; thus, any complete search of C-space would require infinite time. Discretizing the space makes it finite, but still complete search times can, in general, be far too long for real-time. A limited initial search works as follows.

Given a goal position, the algorithm searches the portion of configuration space that satisfies the condition that the robot end-effector, J_5, is at the goal location. For each configuration in the C-space search, the algorithm performs a collision check. If no intersection is detected, the configuration is added to a list of *legal* configurations; otherwise, the configuration is discarded. Because the end-tip location is fixed, the number of collision checks that must be performed is reduced, thus further limiting the calculations required to perform the search.

While the number of steps in the search is, in principle, large, it can be reduced by increasing the step size, or equivalently, reducing the resolution of the search. The required resolution is an engineering consideration and can be adjusted so that the number of computations required can occur in real-time while still permitting a thorough search, even in the context of 3 spatial dimensions; this is particularly true in the context of ground-based control where unlimited computing resources can be made available.

Penalty function

The initial C-space search returns a list of legal goal configurations that can be naturally divided into equivalence classes with the property that, if the planning problem can be solved for one member of the class, it can be solved for all members of the class using essentially the same plan. Thus, for purposes of plan construction, only one member of each class need be examined.

A penalty function assigns a value to each legal configuration and one member of each class is selected having a minimum penalty; this results in a short list of legal configurations. The penalty function is designed so that lower values are more likely to be associated with successful plans. Thus, when the reduced list is prioritized according to penalty function value, the least number of calculations possible, on average, will be required to find a successful plan.

Heuristics to limit C-space search volumes

Most successful plans involve bringing the arm into a configuration in which all parts lie within a disk of radius L_1+L_2, the length of the inner-most link (see

Fig. 1). Thus, plan construction requires moving to, or from, a configuration within the L_1+L_2 disk. Heuristically limited C-space searches are employed to construct these plans.

The primary heuristic is that a legal goal (target) configuration contains information about how to solve the problem, namely, that all one has to do is slide the end tip along the target solution, starting at the joint between links 1 and 2. If the arm were perfectly flexible, this strategy would always work because the target configuration is known to be legal. Given the arm is not flexible, it works exactly if there is sufficient room to manoeuvre. It did not prove possible to find a simple numerical condition characterizing *sufficient room*. Instead, a strategy was worked out employing local experts resulting in a very limited C-space search.

The C-space search is implemented as an iterative process in much the same way that parallel parking is iterative. The number of iterations depends on the amount of space available in which to manoeuvre. The more space available, the fewer iterations are required. Moreover, the less space available, the finer the nature of a move entering into an iteration. Thus, situations where the strategy may fail involve operating the arm in very close proximity to obstacles. Failure to find a plan can also result from a lack of fine motion control, which is another way of saying the resolution of the C-space search is too coarse.

Local experts

Consider the problem of driving a car from your house to a local store. Performing this act requires steering the car, which in turn involves turning the steering wheel to one of an infinity of positions at various times. Thus, in principle, one must decide at each moment in time what the correct position setting for the steering wheel is and set the wheel to that position. One might describe this method as control in local time. For the most part, this is not how one drives to the store. Rather, one performs a series of discrete acts, such as: back the car into the street; turn right; follow the road until ... , etc. While either form of control can work, the latter, which makes use of a series of specific modes, or activities, is clearly more efficient. Moreover, if one wanted to communicate to a visitor how to drive to the store, one would generally formulate a description in terms of these modes, e.g., turn right out of the drive; proceed to the second stop sign; turn left; etc.

Now consider the problem of controlling a 2D-3dof robotic arm. It is evident, subject only to resolution of position control, that any arm motion can be achieved by changing the positions of joints individually, while keeping the remaining joints fixed. On the other hand, one may look for a short list of specific motion types that are sufficient to solve problems. This leads directly to the concept of local experts. Thus, a *local expert*, as used in respect to the PPS, is an algorithm for performing a specific type of arm motion.

For purposes of this discussion, let l_1 (L_1+L_2), l_2 (L_3+L_4), and l_3 (L_5) refer to the three links of the arm from inner to outer as shown in Fig. 1, and let J_i, $i = 0, \ldots, 5$ be the joints shown. Next, let S_i refer to the line defined by the long axis of l_i prior to the initiation of any motion, and T_i be any line through J_i where, S_i and T_i are to be held fixed for the duration of application of each local expert motion. The motion procedures are:

M1: Slide l_3 in one or the other direction along S_3.
M2: Slide J_5 in one or the other direction along T_5 while simultaneously sliding J_4 along T_4. Note, T_5 may be S_3, and T_4 may be S_2.
M3: Rotate l_3 while keeping the spatial position of J_4 fixed.
M4: Rotate l_2 at J_2 while keeping J_5 on S_3.
M5: Rotate l_1 at J_0 while keeping J_5 on S_3.
M6: Rotate l_1 at J_0 while keeping J_4 on S_2.
M7: Rotate l_1 while keeping q_3 and q_5 fixed.
M8: Rotate l_1 while keeping J_5 fixed.

It is to be stressed that each local expert algorithm involves a solution to the inverse kinematics problem. One utility of the local experts approach is that the nature of the solution is constrained so that picking a solution which results in a continuous C-space path, i.e., a *physically* achievable path, is almost trivial.

Via configurations

A solution to a path planning problem is a sequence of configurations, $(q_{1,n}, q_{3,n}, q_{5,n})$ where $0 \le n \le N$, $(q_{1,0}, q_{3,0}, q_{5,0})$ is the initial configuration, $(q_{1,N}, q_{3,N}, q_{5,N}) = (q_{1,g}, q_{3,g}, q_{5,g})$ is the goal configuration, and the intermediate configurations move the arm from the initial to goal in a continuous fashion. Such a sequence is called a *configuration sequence*.

A *via configuration* is a specified sub-goal configuration which forms a part of a configuration sequence. The critical fact here is that if one records each via configuration together with the local expert to be applied to get to the next via configuration in the sequence, the result is a complete set of instructions for solving a path planning problem. This is an exact analogy to the instructions one would give to drive to the store.

Hierarchical organization of planning

The PPS algorithm divides a planning problem into two parts, a global part and a local part as follows.

Consider an initial configuration, $(q_{1,0},q_{3,0},q_{5,0})$, and a goal configuration, $(q_{1,g},q_{3,g},q_{5,g})$. Global planning relates to whether $q_{1,g}$ can be reached from $q_{1,0}$, where the arm could perform a *posture change*, i.e., reverse the sign of q_3, or the conditions under which the arm can be guaranteed to rotate from its initial to goal position without encountering an obstacle.

Local planning involves the use of local experts to perform specific activities. Such are described above, but would also include the facility to develop a posture change plan for a particular spatial location. In general, local planning is focused on a small portion of the operating space, and so the number of obstacles that have to be taken into account is substantially reduced.

Chaining

Chaining, both backwards and forwards, is a primary method for performing the local C-space search. In simplest form, the algorithm starts at an initial configuration and finds a sequence of local expert moves which takes the arm to a configuration in the L_1+L_2 disk. This is a forward search. The algorithm then takes a goal position and tries to convert it to a configuration in the L_1+L_2 disk. This is a backwards search. Finally, the algorithm tries to connect the two intermediates which may involve a posture change.

The use of these ideas is illustrated in Fig. 2. The initial configuration of the arm shows link 3 confined by two obstacles at the right. Forwards chaining is used to find a series of via configurations leading to the configuration labelled 1a-3a which lies completely within the L_1+L_2 disk. A limited discrete search of C-space is performed to find the target configuration. Backwards chaining is then applied to find via configurations that lead from the target to the configuration labelled 1b-3b. Since the angle q_3 (see Fig. 1) has opposite sign between configurations 1a-3a and 1b-3b, the posture must be reversed. This requires a sector devoid of obstacles that is of radius slightly more than the sum of the lengths of the two inner links ($L_1+L_2+L_3+L_3$ as shown in Fig. 1). The position of a suitable sector found by the algorithm is identified by a PR in Fig. 2. Chaining methods are now used to find a sequence of via configurations that will accomplish the posture reversal.

Place in the Literature

Latombe [1991, pp. 11-21] contains a survey of robot motion planning methods. Three main approaches to motion planning are discussed there: roadmaps; cell decomposition; potential fields.

The method presented in this paper most closely fits in as an approximate cell decomposition method, Latombe [1991, p.15]. This description certainly applies to the portion of the algorithm concerned with identifying the free portion of the L_1+L_2 disk and sectors in which posture reversal can be undertaken.

On the other hand, the chaining portions of the algorithm produce sequences of via configurations which are very like road maps, at least in concept. The unique feature of this approach, which is believed to have direct application when dealing with a fully dynamic system, is that the via configurations are linked by specific, simple types of motion by the robot. These are the motions controlled by the local experts. Thus, the issue is not merely that there is a path, but rather that there is a path which can be implemented by physically simple motions of the robot. Although the notion of local experts was introduced in the robot motion planning context by Donald [1987], so far as we have been able to determine, the use of this idea described in this paper is unique.

3. A Path Planning Example

Fig. 3 shows a difficult motion planning problem. The difficulty is not so much in finding a path; rather, because of the placement of the obstacles labelled *x*, *y*, *z* and *w*, the difficulty lies in a user employing a manual arm control method to execute the solution.

In Fig. 3, the initial position of the arm is shown and labelled 1 → 3. In addition, a sequence of via configurations generated by the planning algorithm, 1a → 3a, 1b → 3b, 1c → 3c, and 1d → 3d, is presented from which a target configuration that will solve the problem is obvious. The solution is implemented by getting to 1b → 3b, and then using M_2 to slide J_5 first along 3c (2d) and then along 3d. This solution is easily implemented using the kinematic feature of the planner, which moves the arm along the specified path using motion increments which are small enough to approximate continuity.

The difficulty for a human operator using active control to move the arm to the target configuration arises because of the narrowness of the entry between, and the relative positions on the disk of, the obstacles, x, y and z. To understand the difficulty, imagine the arm moved from its initial configuration (1-3 in Fig. 3), to the via configuration labelled 1a → 3a. Rotating link 1 in a clockwise direction until the via configuration 1b → 3b is reached does not encounter obstacles. However, further clockwise rotation of link 1, to achieve 1c → 3c, cannot be accommodated without simultaneously rotating link 3 in a counterclockwise direction, because to do otherwise would cause link 3 to collide with x. On the other hand, too much counterclockwise rotation on the part of link 3 will cause a collision between link 3 and z. Once the via configuration labelled by 1c → 3c is obtained, further clockwise rotation of link 1 is constrained by possible collisions between link 2 and x, and link 3 with either y or z, depending on whether there is too much or too little counterclockwise rotation of links 2 and 3. In short, solving this problem appears to require incrementing all the joint angles simultaneously at specified relative rates.

Manual Control Solutions

The GTS provides two methods for manually controlling the arm. The first, kinematic control, permits the user to change the angular position value of any of the three real joints by an increment of $\pm 1°$, or $\pm 10°$. The second, dynamic control, permits the user to control simulated actuators which power the joints in the dynamic model.

Now consider the application of the two available control methods to the problem above. Neither method permits simultaneous control of multiple angles at specified relative rates. However, this does not preclude solution to the problem. In the manual mode using kinematic control, the problem can be solved because the minimum increment, $\pm 1°$, is small enough to permit the user to approximate simultaneous incrementing of joint angles by sequential incrementing of joint angles using the minimum increment. Essentially the same solution can be implemented in the dynamic control mode, the primary difference being that the level of skill required is far higher.

Even though the problem in the example can be solved manually, the process is far from satisfactory. Because only one joint is under active control at any time, the task ends up being subdivided into very small pieces, a process which is tedious in the extreme. In kinematic mode, the very tedium of having to divide the task into $\pm 1°$ increments tends to cause frustration which can lead to over estimating the available space, with resultant collisions. On the other hand, in the dynamic mode, trying to implement very fine motions requires significant skill and complete concentration. This again results in frustration and/or fatigue which can lead to collisions.

What is required is a control interface which permits the user to implement complex movements involving simultaneous angle changes at all the joints. A consequence of this assertion is that a good method for specifying the complex motions required is necessary.

4. Application of GTS/PPS to Space Robotics

The focus of the GTS/PPS development projects was to develop technologies of use in furthering the development of space robotics. In the remainder of this section the use of the GTS and PPS as aids in this role are discussed. Specifically, the following potential areas of application are covered: development of methods for supervised autonomous operation and ground-based control of space robots.

Supervised Autonomy

As discussed in the Introduction, prerequisite technologies for supervised autonomous operation and/or ground-based control of space robots have been identified by Jackson [1992] and Immega [1992]. We describe the application of the GTS/PPS system to the development of these technologies.

Task decomposition and trajectory planning

Supervised autonomous operation requires a high level of task decomposition to subdivide a task into parts that are sufficiently simple to accommodate a robot operating in an autonomous mode. As noted, motion plans generated by the PPS have this feature: they are subdivided into segments defined by via configurations with the intervening motions controlled by the application of a single local expert (see Sections 2 and 3). Thus, 3D plans of the type generated by the PPS would already be in the form of scripts suitable for supervised autonomous operation of a space robot such as the SSRMS. To complete the development of this technology, it remains to implement the PPS trajectory generation algorithm in 3D. The task decomposition requirement would be built in.

Trajectory control and error recovery

Motion plans generated by the PPS have two additional useful features:

- dynamic motion control is simplified through the use of local experts;
- error recovery is enhanced.

In respect to dynamic motion (trajectory) control, a high-level control module for a single joint has been designed and implemented as a stand-alone module. While the present scheme focuses on a single joint, because of the special nature of each local expert, it is believed that a complete controller, operating on multiple joints simultaneously, is possible. Schemes for accomplishing dynamic control of multiple joints could simply extend the existing method for a single joint, which has passed limited testing, or employ an enhanced form of a more complicated method suggested by Paul [1981], Section 5.7.

The nature of PPS motion plans also provides for a degree of error recovery. Specifically, as each successive via configuration is reached, the true position/configuration of the arm can be determined from sensor data and compared to the predicted position. If an error is detected, a revised motion plan can be generated. The simplest and most efficient way to recover from such an error would be to revise only the next step of the plan, i.e., revise the plan so that at the next via configuration, the true configuration and position of the arm match the expected configuration and position in the original plan.

Script revision

As a task is being autonomously performed by the robot, there must be the possibility for human intervention and revision of the operating script (Jackson [1992]). Script revision requires an interface that provides the human operator with an effective means for supplying instructions to the robot. In the present case, the focus is on controlling the dynamic motion of a space robot like the SSRMS. In respect to motion of a manipulator, as has been noted (Immega [1992]), the manual control of individual joints is to be avoided. A discussion of some of the reasons why this is the case was presented in Section 3. One way to avoid the manual control of individual joints is to provide an operator with an interface enabling motion control by use of local experts. Preliminary designs for such an interface have been developed that would provide an operator with a graphic tool coupled to a vision system. The key is the local expert concept which defines a specific motion type and supplies the operator with an easy means of specifying the required instructions. Instructions would be of the form: use local expert *X* to move to *Y* (an appropriate form of position/configurationspecification). Once given, these instructions would be converted to a path using the inverse kinematics module for the specified local expert; this path would then be implemented using the appropriate dynamic control module. It is to be stressed that even though the GTS works with a planar 3dof arm, analogous control principles can be implemented in 3D.

Ground-Based Control

The fact that there is a time delay between the issuance of a motion command on the ground and its implementation in space, followed by a further delay between the time a signal detailing an effect is sent from space and that effect is displayed on the ground, is a fundamental difficulty associated with controlling a space arm from the ground. Indeed, experience at dynamically controlling the GTS arm would lead one to believe that fine motion control based on visual feedback in the presence of this type of delay would be impossible. Nevertheless, the GTS could prove a valuable tool in developing an effective ground-based control technology.

While the present GTS does not admit a delay between the issuance of a command and the implementation of its effect, it is straightforward to revise the code to permit this possibility. Thus, a real time simulation like the GTS could be used to study the problems associated with time delays in the implementation of instructions and the visual display of their implementation.

There are other ways the GTS could be useful. These depend on the approach taken to solving the time delay problem, two of which are considered here.

One approach to solving the delay problem is to use a dynamic model to provide a visual estimate of the future state of the arm at the time an instruction issued at the present moment would be effected. A difficulty with this approach, which applies to all uses of models, is that the predicted position/configuration of the arm will deviate from the true position/configuration due to model error. The GTS could be used to study this problem by running two models, one as the prediction model and a second as the 'true' model that differed from the predictive model by the introduction of errors. Errors could be introduced either as a random deviate, or by virtue of using parameter values in the true model which differed from the values in the predictive model. Methods could then be developed to reduce the effect of such errors on a dynamic basis, for example, by using sensor data from the true arm to provide information on how to reduce the error between the predicted and actual position/configuration of the arm.

A second approach to ground control makes use of the type of supervised autonomous operation discussed above. In this instance, as each new via configuration is achieved, the arm is brought to a stop and during this rest period an operator has a chance to determine the true position/configuration of the arm prior to initiating further action, in this case based on a true model which would produce differing arm positions/configurations from the predictive model as described above. This approach would be effective, subject only to an assumption that size of the error grows as a function of the total motion required, because the operator could always intercede using the manual control interface discussed above under script revision. Thus, as the true position/configuration approached a required position, the error generated during the next motion step would be smaller because only a small correction was required.

Both of the approaches suggested above can be implemented and studied using the GTS. Thus, even though motions are confined to a plane, the GTS provides an ability to study the problems associated with time delays and can aid in the development of improved strategies for operator control and monitoring of arm performance.

References

Donald, Bruce R., [1987], "Error Detection and Recovery in Robots," Springer-Verlag, New York, 314pp.

Immega, Guy, [1992] "Advanced Operator Interface for Supervised-Autonomous Operation of Space Robots," 7th CASI Conference on Astronautics, pp. 88-297, November 5-6, 1992, Ottawa, Canada.

Jackson, Eric, [1992] "Ground-Based Control of the MSS," 7th CASI Conference on Astronautics, p. 126, November 5-6, 1992, Ottawa, Canada.

Latombe, Jean-Claude, [1991] "Robot Motion Planning" Kluwer Academic Publishers, Boston, 651pp.

Paul, Richard P., [1981] "Robotic Manipulators: Mathematics, Programming and Control," MIT Press, pp. 1-279, 1981.

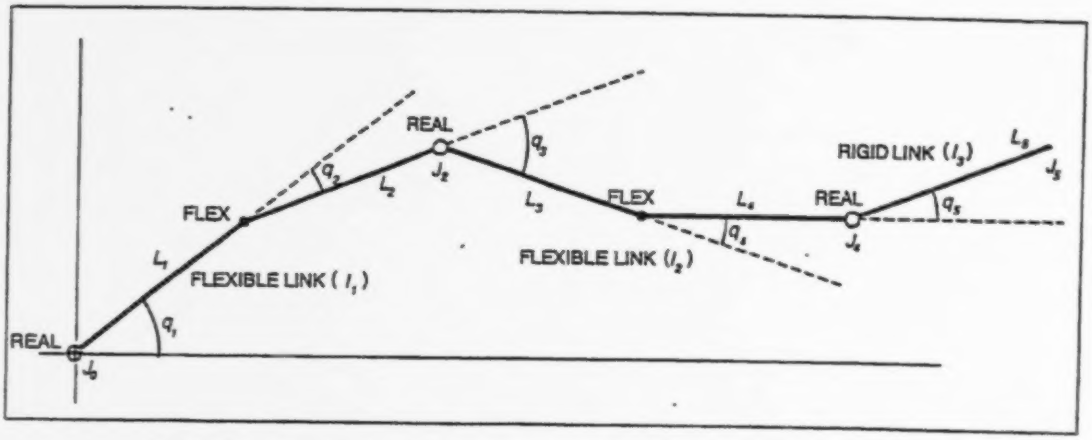


FIGURE 1: The standard configuration and labelling for the 2D-3dof arm with flexibility in the two interior links.

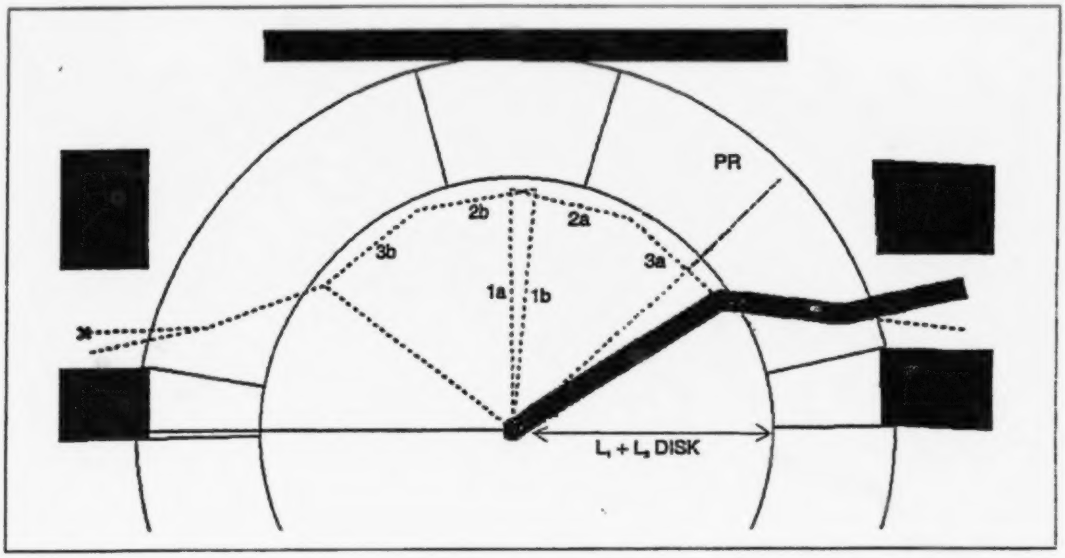

PR
2b
2a
3b
3a
1a
1b
3t
$L_1 + L_3$ DISK

FIGURE 2: A planning example illustrating the path lanning method.

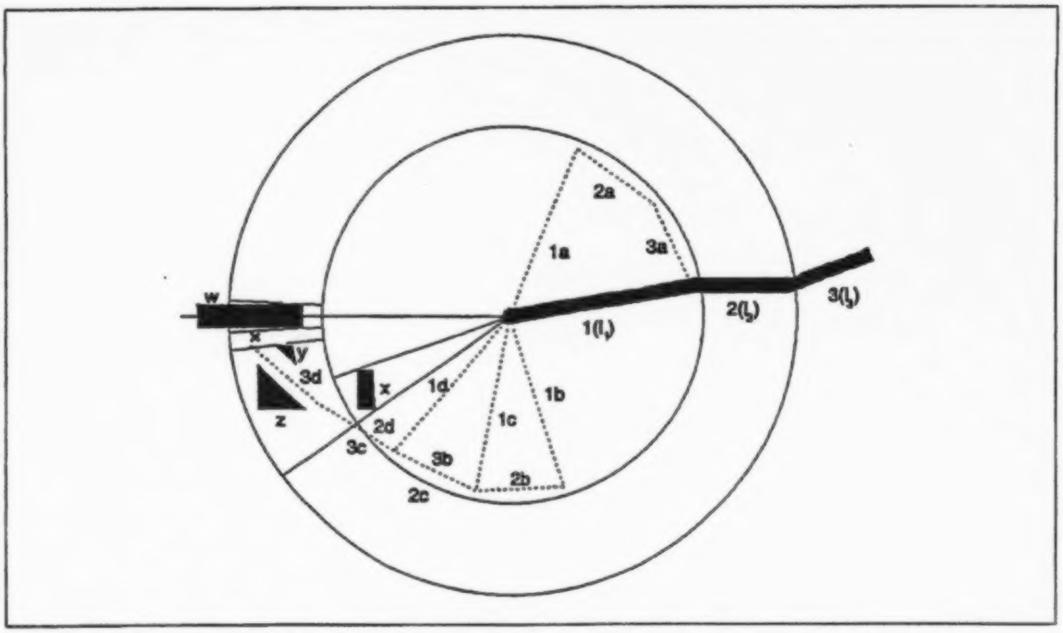

2a
1a
3a
w
x
y
3d
x
1d
1b
z
2d
1c
3c
3b
2b
2c
$1(l_1)$
$2(l_2)$
$3(l_3)$

FIGURE 3: The motion planning problem shown (redrawn from a screen dump) is one of the most difficult because of the narrow entry to the goal which requires very fine levels of control.

243750

Knowledge Technology Infrastructure: Perspectives on the Past and the Future

J. E. Smith
Senior Strategic Planner
National Research Council of Canada

#149637

Abstract

Infrastructures make it possible—even convenient—to apply technology. Infrastructure provides access. Building a competitive knowledge-intensive infrastructure will be expensive and require sustained investment—just like our road system that lets automotive technology contribute to the national well-being. Measuring the success of the new facilities will not be easy at first; we do not yet even know what the appropriate measures are. And the path to national success will be bumpy as we sort out what is most appropriate for the private sector, and what are the responsibilities of governments. The view ahead is not yet clear, but our management of this new infrastructure will determine our economic and competitive welfare, and even impact—hopefully beneficially—on our environment.

Introduction

Good Afternoon. Today I will be talking about technology and more specifically about those technologies that are part of our knowledge technology infrastructure.

We are all familiar with technology. It's principally about ideas, sometimes bad ideas, and about often getting it wrong the first time so that we are forced to learn and improve our techniques and ideas as time goes on. So today I will be focusing on the organization of these ideas and their systemic expression through our knowledge technology infrastructure, because we cannot afford to be always wrong or to take too long to get our systems working correctly and efficiently.

Defining Technology Infrastructure

First we should ask ourselves: What exactly is knowledge/technology infrastructure and what aspects of technology are included? I have several responses:

- One, we should think beyond the familiar hard civic services such as sewers and roads when we consider infrastructure. These familiar elements are important, however the objective of my presentation will be to demonstrate that these physical services are but one aspect of our total investment in infrastructure.

- A second point is that infrastructure is more broadly concerned with technology systems. These systems constitute a foundation for productive activity in an economy—encompassing common ways of communicating, transporting, building, generating knowledge and conveying information, goods and resources.

- Third, infrastructure is essentially about the ways we as a nation organize and distribute knowledge through technology. It determines how our knowledge networks are made accessible for firms, organizations and individuals. Accessibility is perhaps the most important concept for technology infrastructure, since it is what distinguishes infrastructure technologies and systems from proprietary ones. All infrastructure is technology but not all technologies contribute to infrastructure.

- Fourth, infrastructure also describes integrated capabilities that have relatively pervasive socio-economic reach and impacts across a nation, such that they can affect the productivity of the whole society.

- In summary, infrastructure therefore includes more than just physical structures, and this more general concept of knowledge-technology infrastructure is fundamental to our future because it expresses our collective agility as a society, and enables our capabilities for managing innovation processes.

Key Questions for Technological & Infrastructural Assessments

Today we are transfixed by technological change, and we are bewildered by its apparent power, so I will focus on three intriguing questions that are pertinent to infrastructure analysis.

First is a question about economic history and our material existence:

Why are we materially better off than our grandparents were?

This question addresses some basic issues of neoclassical economics and its treatment of technological innovation.

Second I will attempt to describe some of the virtual aspects of our unfolding infrastructure.

What happens when technology begins to dematerialize our lives?

And third, let's look forward and speculate about what might be some of our grandchildren's apprehensions.

What might we say about technology if this was 2093?

These questions remind us that technology and infrastructure are time dependent concepts that are created and shaped, regulated and exploited according to human culture.

The Infrastructure of the Automobile

At the start of the 20th century the first signs of the new culture of internal combustion engines and the automobile were evident, however many of the infrastructure features we take for granted today - from pavement to lane designations were not yet institutionalized. As the automobile based economy evolved, so did production practices, technology components and the institutional and business systems of its supporting infrastructure. Eventually these became the complex relationships that are familiar to us today, and behind each major element are sets of knowledge-technology systems that reflect the accumulated experience, public regulations and market shaping organizations associated with this infrastructure.

Infrastructure change and the dominant technologies that drive it create processes that are somewhat like ecological succession where species gradually and sometimes dramatically succeed each other as surrounding conditions change. Mature infrastructures gradually foster conditions, such as traffic snarl, and high energy usage that are conducive for a successive infrastructure to develop.

In Japan today, for example, one of the driving forces behind their rapid deployment of fibre optics is Tokyo traffic congestion. Now if we take a brief look at the Japanese industrial success story for what it reveals about the composition and design of one's infrastructure, we see that the social organization is fundamental.

Technologies become dominant largely because their supporting infrastructure is broadly accessible and affordable. When this happens they attract intense competition that seeks to apply design and production organization advantages more efficiently to secure market share.

Look how quickly and by what means Japanese producers secured global market advantage:

1. From 1965-1989, Japanese market share in automobiles grew from 4% of world markets to 25% while the US share declined from 48% to 19%;

2. Japan's cost production advantage rose to 20% over the US with a consumer approval rating that was 50% higher;

3. Japan adopted substantial outsourcing well ahead of and at a larger scale than in the US, thus enabling them to become 30% faster at new model introduction and generally in terms of innovation.

4. The key to this social organization is Japan's keiretsu system of interlocking ownerships and shared technology/risk relationships.

The lessons are that by strategic learning and adaptive adjustments to production systems that anticipated the infrastructure changes resulting from the energy crisis of 1973, Japanese producers achieved significant cost and quality advantages. In particular the vertical keiretsu systems of suppliers and assemblers were instrumental in this success, largely because they were designed to share knowledge and innovation within their infrastructure networks to a greater extent than in the US.

Importance of Infrastructure

Several additional points can be made to reinforce the relevance and importance of infrastructure.

The organizational interconnections seem to be critical for successful infrastructue development. Firms rarely act alone to achieve technological dominance. Second, within nations, as Michael Porter has documented, internal technological competition provides a discipline that, when supported by shared infrastructure goals and norms, fosters the continuous upgrading that is the key to competitiveness.

Third, all technology, and hence infrastructure, is knowledge dependent. Even concrete and pavement formulas are continually being refined at NRC to seek better weather resistance, and new epoxies for bridge repairs and truck loading designs are being tested. At NRC we have recently created a new Center for Surface Transportation Technology that consolidates our R&D in this area, and we are also considering the development of a the first comprehensive urban infrastructure code in North America.

Finally, infrastructure is not homogeneous throughout the world: it is particular to local conditions and needs, but increasingly the differences which characterize nations in this regard are being overcome by global technological integration and interdependency. This is largely because the basis for wealth has shifted from resources to technology, and with this shift, the less wealthy want to acquire and replicate the infrastructure design elements of the wealthy nations.

Technology, Infrastructure and Economics

Now let's return to the relationship between technology and economics. Some of the best contemporary analysis of these relationships has been done by Dr. Richard Lipsey of the Canadian Institute for Advanced Research. Many of you may remember his name from your introductory texts in microeconomics.

Dr. Lipsey is now suggesting that technology warrants inclusion as a third defining variable for economics, along with capital and labour, in a new dynamic equilibrium structure for economic theory. In this formulation, technology represents an endogenous variable that embodies the value of knowledge and new ideas within a culture. This gets us away from static equilibrium into a new structure where ideas are critical assets that must be managed effectively to create new value by stimulating innovation and economic growth.

Viewed in this context, infrastructure can be regarded as an important capital investment even when this investment is knowledge-based as well as just physical in nature. As a result, intangible assets can now be factored into economics more readily.

Why are we materially better off than our grandparents? This is Lipsey's question. Because of new inventions that are derived from our technological capacities, and our institutional organizing systems. Market systems create the demand structure for these innovations and infrastructure helps to shape and determine their broad application.

Technology and infrastructure are also symbiotic in that each is necessary for each other. For example, new digital broad band technologies both enable a new communications infrastructure and create the basis for markets in specific services that previously were not possible or widely accessible. And finally, we retain infrastructure elements from the past which are still useful, adapt those which are suitable and replace those which fail to add value in the new scheme of things. The Erie Canal, a mainstay of 19th century American commerce is now a tourism asset; the telephone is no longer just a voice communicator.

Central to the new economics of knowledge and technology is a typology of innovation that distinguishes four types: incremental, radical, clustered innovations and fundamental or comprehensive paradigm shifts. Dr. Lipsey believes, and I agree, that since 1980, western culture has been entering a period of profound paradigm shift that is now moving us rapidly out of the industrial era and into the knowledge era.

Three Types of Infrastructure

Now I want to switch back to our typology of infrastructure. At NRC when we talk infrastructure we are including three levels of activity. Think of these as inputs, throughputs and outputs if you like, but not necessarily in a linear or repetitive sequential way because infrastructure functions tend to be more like the random access memory of your computer than a hierarchy of steps.

The first level is science and technology dominated, and this is where NRC has its greatest investment. These are the infratechnologies, of R&D, the informational software elements that are associated with the production of new knowledge that will be broadly applied to our systems of innovation within firms and across the society. Codes and standards knowledge and measurement, diagnostics are good examples. Modern, complex societies rely upon their knowledge bases for continuous refinements and new breakthroughs, or at least reasonable copies of the advancements of others. This knowledge base, in current parlance, forms a type of input software for the development of the other levels of infrastructure.

A second level is that of the knowledge exchangers, modifiers and adopters—primarily ongoing and virtual institutions from the enduring to the temporary. This includes universities, networks, programs, advisory boards etc.—the institutional organizations that matter. These throughput mechanisms or infomediaries play critical roles as surrogates for the market before it knows what it wants or will accept. Here is where the infrastructure pulls and pushes to find dynamic technology entry points and where much innovation gets shaped. Even this conference, is a potentially important infrastructure infomediary in establishing a new shared vision of what options our future may hold.

The third level is more tangible, consisting of the outputs, hardware and physical technologies that we typically associate with infrastructure: civic services, telecommunications and transportation systems. What is amazing now however is that even here, the balance is shifting toward the intangible, knowledge based system components. What most observers miss is the changing traffic loads and value relativities, as virtual elements get substituted for physical ones.

To briefly recap: knowledge/technology infrastructure is where private and public innovation discover what will hold as significant for the next economy during particular time periods. When these periods coincide with major paradigm shifts, around once a century, the throughputs, rates of change and pressures on all three levels are transformative, even revolutionary.

Infrastructure is a distributive concept: it promotes equity across sectors and industries, and within nations as governments seek to ensure wide access to critical technologies and the prospects for wealth creation.

Finally, building a competitive infrastructure, particularly a knowledge-intensive one is very expensive. Sustained investments are required at all levels, especially for diffusion of knowledge and movement of technologies along the innovation chain so that they can contribute to national competitiveness.

Paradigm Shift

Now I will return to the context of the techno-economic paradigm shift to look at it from an infrastructure perspective. The table below describes the world as we have known it, and largely as our parents and grandparents knew it, accounting for incremental, radical and clusters of innovations that have adjusted and updated the industrial base of the economy that began in the early 1800s.

While the infrastructure elements that supported this system have gradually become more efficient, the structure itself remains resource intensive and mass dependent.

Traffic hierarchies for our infrastructure continued to be materially dominant. Virtually all of the infrastructure characteristics of this economy—transport, utilities and communications have been premised upon usage patterns associated with mass homogeneity. Consequently this infrastructure is relatively inflexible, slow to respond to demographic changes, and relies upon quantity measures to determine its performance.

Now let us examine some of the potential structural characteristics associated with the emerging paradigm of a knowledge-based economy, which I have called the cyber-knowledge economy because of the importance of computerized systems. As Nuala Beck and other economic observers have advocated, it's hard to know where you are in a new economy by measuring performance with old indicators, so first we have to try to understand what some of these new factors might be.

We can't be certain about transitions on this scale, but many regard the accessibility and geometric power increases in personal computing of the 1980s as where the paradigm divergence really began. In the emerging cyber/knowledge economy, information becomes more productive than goods, the base for commerce and infrastructure traffic shifts toward intangibles, and older infrastructure elements get smarter.

I do not wish to sound like a total technology optimist or determinist, so I think its worth noting that we don't yet know what most of the implications of these shifts will be. Whether *real knowledge* and its reflective, cynical alter ego, *wisdom*, also will be preserved or increased is a challenge probably for our children to grapple with. We are still at an early stage with information as a primary vaue driver, not unlike Thomas Edison before he found a way to meter electricity.

The shape of the emerging paradigm that we can see has a much more intangible character, and relies upon a different materials base than before. It also is developing capabilities for customized service, design and delivery that rely upon programming power rather than mass systems. This in turn creates new infrastructure forms such as digital networks, multimedia and internets that are not only very agile and changeable, but are also being pushed ahead more by fairly small, agile firms than by large scale institutions that remain wedded to the old paradigm. The result may be a more dematerialized world; economies of scope rather than scale.

Implications of Paradigm Shift

What does it all mean for you or me? If you are an auto parts manufacturer, it may mean that you must anticipate the new infrastructure by informationalizing your business or technology before it goes the way of the steam engine. Instead of trying to develop a better corrosion resistant steel for mufflers, you would probably want to investugate noise cancellation software and durable microphones attached to programmable noise control chips.

There will also be new technology fusions that combine old fields in new ways: such as electronics and optics to make optoelectronics, or graphics and molecular structures as bioimaging. Parts of the infrastructure will find new interdependencies, like smart highways and global positioning systems.
We can also expect to see a shift in economic externalities from pollution to privacy.

Trends in Infrastructure

Five trends in infrastructure are particularly apparent today:

1. Privatization of knowledge
2. *Proliferation of networks*
3. Democratization of networked information
4. Miniaturization, integration, convergence
5. Internalization of environmental factors.

The increased involvement by private firms in the development of infrastructure technologies, whether it's toll roads or broad band networks means that knowledge is being privatized for strategic periods of time before its accessibility is assured.

There are more channels and network options for communicating similar messages than ever before. Broadcasting is giving way to a new digital smorgasbord of faxes, desktop publishing, e-mail, personal electronic news and video, soon to be integrated in early multimedia networks. From freenets to groupware, software advances and newtwork architectures have created new freedoms and anonomous communities for communication and brainstorming. And the Internet has become a surfer's paradise with few regulations or policing, reflecting the fundamental democracy and anarchy of networks design.

Miniaturization took hold in the 80s and has continued unabated. Computers have, in two decades gone from rooms to palmtops, and utility power is now starting to migrate away from central plants to site based cogeneration, fuel cells and photovoltaics systems. Next we will see the advent of substantial arrays of really miniature machines or nanotechnologies for tasks we can hardly begin to envision now.

Finally, I believe that we have to recognize the internalization of environmental factors. Environment becomes just a new norm to be respected and maximized in the value chain by the application of the best knowledge available. Most of these trends will improve the environment because they require fewer resources, ie. they are de-materializing, and they enable environmental knowledge to be much more easily incorporated into our productive processes.

So what are the downsides, the things that can and will go wrong as we move into the new paradigm? There are several critical issues that most analysts believe must be resolved before the new infrastructure conveys seamless services in accordance with projected potential.

These are:

1. Ownership and access;
2. Mobility management;
3. Interactivity protocols;
4. Security and encryption;
5. Real time connection and selection;
6. Interconnectivity standards;
7. User/receptor authentication
8. Investment alliances(public-private)
9. *Interference and overload.*

I believe that the most critical of these concern encryption issues that will determine how information metering can be accomplished, and what will remain of privacy rights in what will become a most pervasive and perhaps invasive infrastructure. These issues are now the subject of much debate as the US begins to put in place its legal and investment foundations for the National Information Infrastructure.

Smart Infrastructure Networks

Before we tackle the imaginative horizons of the next century, I think it's worth some time to review briefly the current array of emerging new infrastructure components. Here is a list of the anacronyms which you will hear more about if they are not already familiar.

Smartcards, HDTV, Biomedical Imaging, Edison Project, Iridium, ISDN, NII, NREN, CANARIE, FNA.

Iridium, for example, is Motorola's satellite based wireless and wired, cellular and regular seamless system for global data, voice and image transmission. Locally, the Stentor/Newbridge financial network association (FNA) has great potential for Canada and Ottawa as global financial transactions seek appropriate hubs where infrastructure quality, service and upgrading can be assured.

On the physical side we are moving rather rapidly to adjust our conventional systems to become smarter, smaller, faster and more user responsive.

Some of the smart physical systems now appearing include:

IVHS: Intelligent Vehicle Highway System;
TGV high speed rail;
GPS global positioning systems
Earth Observation Satellite System
Fuel cells, Photovoltaics, Hydrogen energy systems;
DSM demand side management for utilities
Intelligent buildings
Automated flight control.

The Distant Future: Looking Toward 2093

In my final series of slides I want to return to the future as seen through present science fiction, where the current reality is more virtual than tangible. I believe that we have already begun to stumble into cyberspace, hesitantly but inextricably so. Cyberspace is a concept popularized by William Gibson in his sci-fi novel Neuromancer in 1984. Cyberspace remains a somewhat elusive concept, but one that is gaining currency as people attempt to understand the paradigm shift. Cyberspace has been described in many ways. Here are a few: data abstracted from computer banks; a netherworld of non-space and transactional, invasive relationships that center on a future world of cyber-bio manipulation.

This is mainly darkness and scattered light but it presents for us a challenge for our evolutionary conciousness of science and technology and the future, much as H.G. Wells, Asimov or Da Vinci have in the past. There is clear trepidation we must feel as we consider these types of future possibilities.

If this virtual world is to be a master matrix of data, are there any smudges in the light lines? Do these constellations of data mean anything profound for our species or will they be just different types of vehicles for accessing the infrastructure functions that we will still require to exist? Will we really be different? Many think we will be as different in 2093 from now as we are compared to our forefathers who preceeded the industrial revolution.

While cyberspace remains a mysterious and still threatening concept, virtual reality seems more friendly because it is more clearly an extension of the present. VR, as it is known by is a multisensual tool that can mediate in relationships, be it conversations, business, sex or learning. You can even visit the National Gallery today to see an early demonstration of this technology. Artists are having fun with it.

VR is the next phase in shifting our computing focus from objects to being. Since the 1950s, each decade has removed a barrier between the computer and the user: Away went the computer room, the time sharing and the batch cards, the rigid menus, and now the keyboard.

Next we are going through the looking glass and into the machine. In history, these types of changes, with writing, printing, and more recently word processing, copying and faxing, lead to changes in how we regard ourselves and our capabilities. We tend to intensify our knowledge searching as we seek coexistence between our experience and our cyber capabilities.

I now conclude this presentation by returning to that question asked near the beginning: *What might we say if this was 2093?*

From inside or outside cyberspace and embracing both reality and virtuality, we will be led back to the enduring aspects of our historical evolution as a species. First we will be still wondering who we are and why we are here. Our search for extraterrestial life will still be central to our ontological dilemma. Second, we hope that our relationship with GAIA—the planet regarded as a living organism—may be more fully resolved with mutually sustainable confidence. Third, the biotechnological and genetics issues of the 21st century will have exerted strong challenges and perhaps some resolutions that will either bring new vistas or greater confusion. And finally, I agree with William Gibson that the essential state of the 21st century techno-economy and infrastructure development will be one of constant apprehension.

I leave you with a question that neither you nor I can answer:

Will we adapt to exist in the violent and manipulative world of William Gibson's Neuromancer, or will we discover a new romance with technology that is both friendly and sustainable?

Thank you.

9OC 223

Learning Machines based on the Evolving Transformation Model

Vithal N. Kamat, Virendra C. Bhavsar and Lev Goldfarb

Faculty of Computer Science
University of New Brunswick
Fredericton, N.B., Canada, E3B 5A3.
Ph. (506) 453-4566, Fax (506) 453-3566
E-mail: u095@unb.ca, bhavsar@unb.ca, goldfarb@unb.ca

#149C38

Abstract.[1]

This paper describes the operation of learning machines which are based on the Evolving Transformation Model (ETM). This model unifies the two basic models used for learning, namely the vector space or geometric model (e.g. Connectionist Networks (CN)) which are numeric (analog), and the syntactic or structural model (e.g. Symbol Processing Systems (SPS)) which are discrete. The ETM thereby possesses the advantages of both the CN and SPS models. A distance function forms the basis for the unification of the CN and SPS models.

We present a learning machine which is suited for handling input patterns coded as tree structures. The capabilities of the machine are demonstrated using handwritten characters recognition problem. When a machine is presented with a collection of different handwritings of the same character, the machine finds a characteristic concept which enables it to distinguish members of this class from others. The concept description is obtained in the symbolic form, although the distance mechanisms are numeric in nature. We also show that when the machine is given a set of handwritings belonging to different individuals, the machine is able to carry out identification of the characters of the different individuals. We believe that the ETM based learning machines have enormous potential for many pattern recognition applications.

1. Introduction.

The ultimate aim of developing a learning machine is to enable humans to indirectly perform tasks in unfriendly environments. The environments could be previously unknown to man or to the machine, thus eliminating the possibility of preprogramming. The environments which fall under this category range from space exploration to mining chutes to ocean bottoms. The learning machines can also be used to handle many of the tasks performed by the present day computers more efficiently, e.g. handwritten character recognition, data management and missile tracking.

The main computational approaches to learning machines are the Connectionist Networks(CN) and the Symbol Processing Systems (SPS). Since CN are only numeric (analog) while SPS are only discrete models, both of these approaches are inadequate for building learning machines which can perform complex tasks described above. The inadequacy of the syntactic (symbolic) approach is obvious. It is difficult for a symbolic system to handle *noisy* real world applications with *pure* concepts from automata and the theory of formal languages. On the other hand a CN process essentially represents only a weight adjustment process. The absence of a framework for constructing new features in the course of learning makes CN inadequate for learning.

In order to arrive at a satisfactory learning model, we have to first formulate the learning problem. *Inductive learning* provides for the most challenging category of *thought* for modeling. *Induction*, on which inductive learning is based, is a systematic way of reasoning, that increases the given information[4]. As described by Watanabe[12], induction is a process of producing a rule or a general proposition, based on a set of facts or particular propositions, such that the facts become special instances of the former. Induction therefore describes a generalization process, i.e. a process of going from particular to general (universal).

In the next section, we give a correct formulation of the learning problem based on such an inductive generalization. In Section 3 we explain the role of distance and show its generalization. We present an *Evolving Transformation Model* (*ETM*) as our learning model in Section 4. We concentrate specifically on the Evolving Tree Transformation

---

[1] This work is supported by Canadian Commonwealth Scholarship, CBIE- grant # KAMAT 91-145-292.

System (*ETTS*) since we represent our input objects as tree structures. The ETTS approach is unique since it combines for the first time, both the conventional approaches, namely CN and SPS. We next explain the learning process using the ETTS model in Section 5. The capabilities of a learning machine based on ETTS are then demonstrated with the aid of a handwritten character recognition problem in Section 6.

2. Learning Problem

In order to define the learning problem, let us consider the following example. Given a *Learning Machine* (*LM*) which has not seen any automobiles earlier, we wish to learn and recognize cars. In the training phase we show in one class a small set of cars while in another class a set containing a few automobiles which are not cars. Let us call the former set the positive class and associate the label 'Car' to it. One of the goals of the LM is to extract the concept associated with this (positive) class of cars (e.g. carries passengers, has four wheels, etc), which allows us to identify it from any automobile in the other (negative) class. With this concept learnt it should be possible to identify any other car which was *a priori* unknown to the LM. Next, we may wish to also generate subclasses in this positive class. For eg. we may wish to know if a specific car is a sports car, a convertible, a hatchback or a sedan, etc.

The prime requirement of inductive learning is, therefore, the ability to generalize. Generalization allows us to map a large number of instances into a single concept. The objective of generalization is to achieve optimum storage of learnt information.

Also, for systematic information storage and retrieval, and for fast access times, we need a hierarchical structure. For this we require a mechanism by which we can split a class into subclasses, on the basis of some features possessed by the class objects. These two requirements form the basis for our learning problem which can be stated as follows:

Given two finite sets, C^+ and C^- which represent the set of positive and negative training objects respectively, the objective is to learn a concept C which belongs to every object of C^+ and to none of the objects of C^-. An additional objective is to identify the set of features F in C^+ that are characteristics making the objects of C^+ look dissimilar to each other; these features enable subclassification.

The set C^+ represents a small subset of a possibly infinite set which possesses the same concept C. Since the concept C is a unique property of the objects of C^+, its discovery enables us to recognize any object which contains C, including those which do not belong to C^+. The recognition problem is therefore a trivial one, and can be stated as follows.

Having discovered a set of concepts and its one to one correspondence with the training sets, the objective is to check if a new object (*a priori* unknown to the LM) possesses a concept which match to any of those discovered, thereby enabling the new object to be classified.

Therefore, we can observe from the above discussion that the major task is learning and discovering the concept C. Recognition is a simple classification problem. We can also note that the features F have no role to play during recognition if only class C^+ membership is to be checked. However, F aids subclassification of the positive class, thereby enabling hierarchical categorization of the objects.

3. Role of Distance

The underlying concept of the ETTS is the distance. As postulated in [1], the distance function is the concept which unifies the symbolic space and numeric (vector) space models.

Levenshtein[8] was the first to introduce the concept of distance (γ), as the smallest number of insertion/ deletions (indels) and substitutions required to transform a sequence *A* into another sequence *B*. Weights were later introduced in order to make some operations costlier than others. Association of weights to the labels enables edit operations on nodes with certain labels to be made costlier than others. The need to generalize distances, was then felt in order to simulate biological mechanisms, where a single action could result in the indel of an entire consecutive string into a macromolecule. Gotoh[3] used the method known as "linear gap weights" to compare macromolecular sequences. Kruskal and Sankoff [7] show a similar but more general method based on generalizing Levenshtein distances. They not only permit weights to be associated with individual indels and substitutions; but additionally, weights to be associated with strings of consecutive insertions or consecutive deletions. Such a scheme allows edits of entire sequence of consecutive nodes in one operation at a cost lower than the sum total cost of editing each node individually. The edit operations therefore consists of single node operations (*elementary* operations), and also chunked edit operations (*macro* operations) where every chunk consists of consecutively linked set of nodes. The elementary and macro edit operations together are termed as *Generalized* (*G*) operations. There is also a discussion of generalized string distance by Santoso[11].

We have extended this generalization concept further to trees. Tree structures provide versatility in representation of complex structures in comparison to strings. Therefore we developed a tree distance algorithm which can accept both simple elementary operations as well as larger macro operations. For this we have adapted and extended Zhang and Shasha's tree distance algorithm[13] to handle elementary as well as macro edit operations. We have given a detailed presentation of the generalized (G) operation based tree distance algorithm in [5] and [6].

4. Evolving Tree Transformation System

The *Evolving Tree Transformation System* (*ETTS*) model is our adaptation of the Evolving Transformation System (ETS) described in [2] to trees. Its evolving or dynamic nature makes it superior to a formal grammar model which is static.

ETTS at any particular iteration t, is a three tuple:

$$ETTS(t) = \{T, P(t), D(t)\}$$

where t = 0, 1, 2, ..., are time intervals,

T $= \{T_1, T_2, T_3, ...\}$ is a set of trees over some finite alphabet A,

$P(t) = \{p_i(t)\}_{i=1}^{m}$ is a finite set of m production rules at the particular iteration t, and

$D(t)$ is a family of distance functions defined on the set of trees T, at iteration t.

The finite set of production rules $P(t)$ refer to a set of m reversible substitution operations $S(t) = \{s_i(t)\}_{i=1}^{m}$, which can transform one tree into another. The production rules also specify a fixed set of composition (*C*) rules which generates the substitution operations $S(t)$ from its earlier operations $S(t-1)$. Also, each of the substitution operation $s_i(t)$ is associated with a weight $w_i(t)$. At an iteration or stage t, if we assume m substitution operations, then the weight parameter set $\omega(t)$ can be defined on an (m-1) dimensional unit simplex $\Omega(t)$ as

$$\Omega(t) = \{\omega(t) = (w_1(t), w_2(t), \ldots, w_m(t)) | w_i(t) \geq 0, \sum_{i=1}^{m} w_i(t) = 1\}.$$

The competing family of distance functions $D(t)$, plays a key role in ETTS. This is a 'parametric family of distance functions' defined on the set of trees T (also see [1]). Each distance function $\Delta_{\omega(t)}(T_1, T_2)$, from the competing family of distance functions $D(t)$ between two trees T_1 and T_2 is formally defined as follows:

$\Delta_{\omega(t)}(T_1, T_2) = \min\{ \gamma(t, \Psi_{12}) \}$ where, Ψ_{12} is an edit operation sequence taking T_1 to T_2.

$\gamma(t, \Psi_{12})$ represents the distance between the two trees T_1 and T_2. In other words it represents the least cost edit operation sequence which the ETTS selects from a large number of sequences. This it does by adapting to the changes (increase) in the number of substitution operations, and also to the changes in the associated weights with every stage or iteration. The competing family of distance functions is defined as:

$$D(t) = \{\Delta_\omega(t)\}, \quad \omega(t) \in \Omega$$

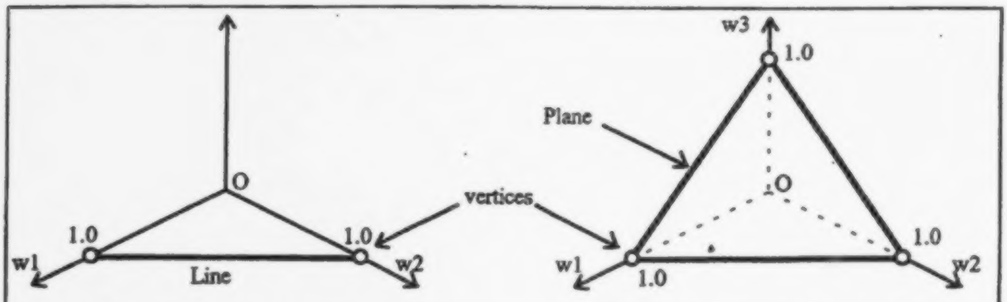
The ETTS defined above generates a more optimal set of production rules and distance functions with every iteration. Therefore, unlike formal grammars, ETTS is capable of evolving with time. If the substitution operations $S(t)$ and the weights $w(t)_i$ remain constant through every stage (i.e. $S(t) = S(t-1)$ and $w(t)_i = w(t-1)_i \; \forall \; (0 < t \leq m)$), then the distance corresponds to its simplest form namely the Levenshtein Distance as follows:

$\Delta(T_1, T_2) = \min\{ \gamma(\Psi_{12}) \}$, where Ψ_{12} is an edit operation sequence taking T_1 to T_2.

The C rules refer to a fixed set of simple concatenation rules to construct new substitution operations $S(t)$ from the existing set of substitution operations $S(t-1)$. One example of such a rule would be to specify how to build a new concatenated operation $ab \leftrightarrow \Lambda$ from previous operations $a \leftrightarrow \Lambda$ and $b \leftrightarrow \Lambda$. Here, Λ represents null and $a \leftrightarrow \Lambda$ represents either $a \leftarrow \Lambda$ (insertion) or $a \rightarrow \Lambda$ (deletion) operation. Building a new concatenated operation or a *macro* operation is similar to the chunking mechanism which is present in intelligent living organisms. Miller[9] showed that human beings can store seven chunks of information in their short term memory regardless of its size. Just as bigger and bigger chunks are built from smaller ones, larger and larger macro operations are built starting from the elementary one symbol operations in our machine. A detailed exposure on macros and their operations in the tree environment is given in [5], [6]. Having formally defined ETTS, we proceed to show its operation and properties in next section and an example illustrating learning with ETTS in Section 6.

5. Learning with ETTS

Let us split the learning process into two distinct stages. In the first stage, we concentrate on the



(a) Unit simplex (line) with two vertices (b) Unit simplex (plane) with three vertices

Fig. 1. Unit simplex showing the dynamic structure change

extraction of the concept C, while in the second, we extract the feature F. Depending on the needs of these two stages we have to set the parameters of the ETTS.

Let us consider the concept extraction stage. We need to identify and set the composition rules which would enable us to generate the new set of substitution operations at every instant of time. For concept extraction process, a single composition rule is sufficient to guide the generation of new $S(t)$, which can be stated as follows:

All possible juxtaposition of the symbols of the alphabet with the earlier generated set of operations $S(t-1)$, results a new set of operations $S'(t)$. The subset of $S'(t)$ consisting of operations which are present in the same number of occasions in each of the objects of the class, gives the set of new operations $S(t)$ •

This rule describes the method of expanding the macro operation at every iteration of the learning cycle. The starting point is the basic set of elementary operations or the symbols of the alphabet. Not every macro operation expanded at a certain iteration are used in the next iteration. For this we use distance functions to provide the necessary guidance in the selection of macros after every iteration (including the starting one).

Let us now consider the feature extraction stage. In this case, we are interested in any macro as long as it is present in atleast one of the objects of the class. Hence, the composition rule is slightly modified as follows:

All possible juxtaposition of the symbols of the alphabet with the earlier generated set of operations $S(t-1)$, results a new set of operations $S'(t)$. The subset of this set $S'(t)$ consisting of operations which are present in at least one occasion in *any* of the objects of the class, gives the set of new operations $S(t)$. •

The addition of macro operations to the set of basic operations brings in a dynamic nature to the learning algorithm which can be best understood with the aid of a simplex described in the following section.

5.1. Unit simplex

Let us assume that we have an alphabet A consisting of two symbols, say $A = \{a, b\}$. At the beginning, say time t_0, we have two substitution operations $S(t_0) = \{a \leftrightarrow \Lambda, b \leftrightarrow \Lambda\}$, which correspond to the elementary operations. To each of these operations is assigned a non zero weight $w(t)_i$, where $i = 1, 2$. For convenience we consider the per unit system where the sum of weights assigned to all the operations at any particular instant of time t equals unity. For m substitution operations, the weight parameter set $\omega(t)$ is defined on a m-1 dimensional unit simplex. For our example with two substitution operations, Fig. 1(a) shows the one dimensional simplex which is highlighted as a bold line. As we move along the line we cover the entire range of weight distribution from $\{w_1, w_2\} = \{1.0, 0\}$ to $\{w_1, w_2\} = \{0, 1.0\}$. In order to understand the dynamic nature of this unit simplex let us consider at the next instant of time t_1, the addition of a new macro operation say ab, i.e. $S(t_1) = \{a \leftrightarrow \Lambda, b \leftrightarrow \Lambda, ab \leftrightarrow \Lambda\}$. The unit simplex now changes its structure from a line in the earlier case to a plane as shown in Fig. 1(b). The weight w_3 corresponds to this new macro addition. The weight distribution could lie anywhere on the plane bounded by the edges joining the three vertices, and inclusive of the vertices themselves.

This explains the dynamic nature of the simplex. However, we also need a weight distribution scheme which would suit the dynamic updates of the substitution operation. For this we define the parametric family of distance functions which allow the various substitution operations to 'compete' with each other during weight assignment (see also [1]). The weights adjustment are so performed, that it allows learning of the positive class. In other words it gives the best separation of the positive class from the negative ones.

5.2. Optimization Function

The concept and feature learning process are similar in that, both can be expressed as optimization processes. The objective of the learning processes is to make the objects of the positive class to look similar to each other and at the same time different from the objects of the rest of the environment. Since we use distance as the dissimilarity measure, the problem therefore reduces to distance optimization.

Let $f_1(\omega)$ be the $\Delta_\omega(t)$ distance between the objects of the positive and negative training class at the center of gravity (*cg*) of the simplex. The distance $f_1(\omega)$ shall also be referred to as the interclass distance. Let $f_2(\omega)$ denote the average $\Delta_\omega(t)$ distance between any two objects of the positive training class at the center of gravity (*cg*) of the simplex. We shall also refer to $f_2(\omega)$ as the intra positive class distance. Then the goal of the optimization process is to obtain a large $f_1(\omega)$ and a small $f_2(\omega)$. Simple weight adjustment may not be sufficient to obtain the desired values of $f_1(\omega)$ and $f_2(\omega)$. In such cases we may also need addition of new operations. Therefore learning is achieved through a feedback controlled process, which adjusts weights as well as adds new operations. As we can observe from the simplex described earlier, addition of new macro operations adds a new dimension to the simplex. The weight vector is then selected on this new simplex, which gives the best separation of the classes.

6. Handwritten Character Learning Example

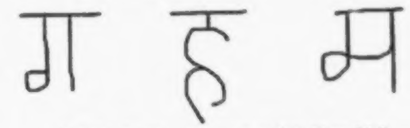
We demonstrate learning with the help of a handwritten character recognition problem. Since these characters emphasize boundaries, it is easy to extract the skeletal structure. Also the degree of randomness of handwritten characters make them suitable for a learning example. We choose the characters of the Devanagari script from India for our example. We preprocess these Devanagari characters and encode them into labeled trees as described in the following section.

6.1. Preprocessing

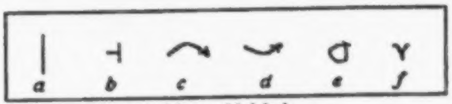
Preprocessing refers to segmentation or decomposition of the pattern into subpatterns. A stagewise decomposition may be performed where the pattern is systematically split into smaller and smaller patterns, until the whole pattern can be fully represented in terms of a set of pattern primitives. Each pattern primitive type is associated with a syntactic operation which is referred to by a label.

Simple as it may seem, such a preprocessing is very vital to the performance of the learning system and therefore should be performed carefully. The form of encoding should be such as to preserve the characteristics of the pattern structure.

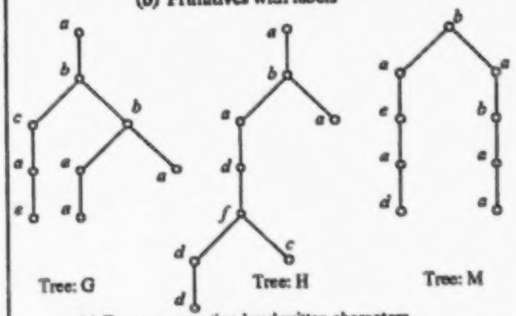
Therefore we need a system which does not give equal stress to the straight fragments and the curved complex fragments of the pattern (character), unlike Freeman's chain coding. Muchnik[10] describes a learning machine which uses for its vocabulary two types of words representing respectively, the *form* and *location* of the characteristic fragments on the image. In his paper he presents an algorithm and a hypothesis on 'pattern of fixation of eyes when a person looks at an image'. We shall proceed with the encoding of our examples in a manner similar to Muchnik, except for a few modifications. Instead of extracting only portions of the image corresponding to the eye fixation points, the method adopted here encodes every portion of the image. The primitives are so selected that the image sections with more information content (change in slope or curvature) can be fragmented more closely in comparison to image sections with less information content.



(a) Handwritten Characters (G, H and M)



(b) Primitives with labels



(c) Trees representing handwritten characters

Fig. 2. Devanagari handwritten characters and their tree representation.

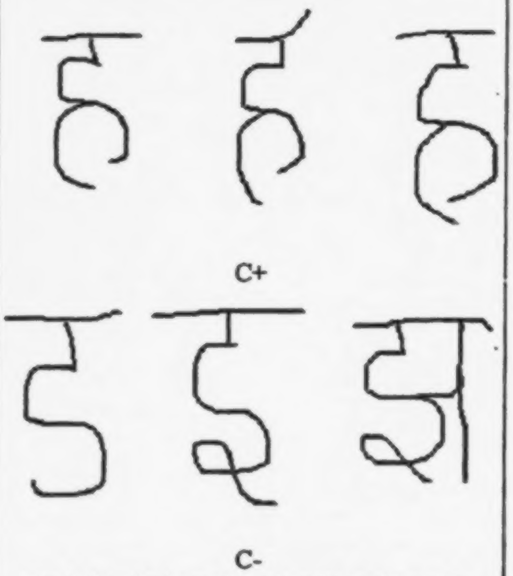
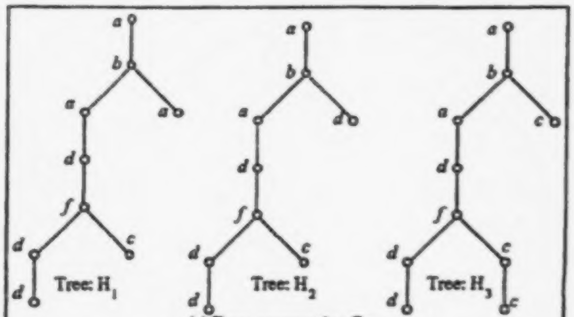


Fig. 3. Handwritten characters (a) Positive class (C+) and (b) Negative class (C-).

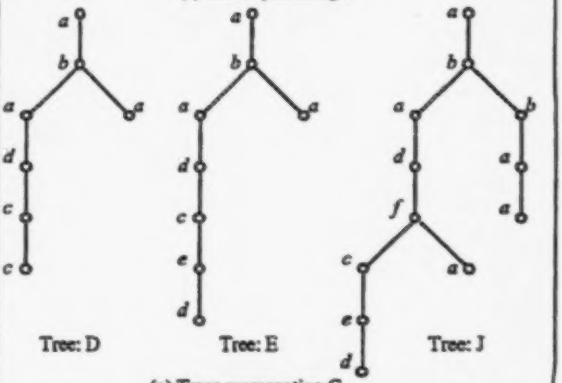
Accordingly the curved primitives are smaller than straight ones. Since we are interested in a tree structure we have to break closed loops if any. Also, consistent encoding amongst all the class characters have to be ensured, so that the root nodes of the trees represent the same start position on the characters. We adopt the convention to start encoding from the top left hand corner of the handwritten character. Then the first node encoded with the aid of the given set of primitives shall be considered as the root node.

For the Devanagari character example we have selected six primitives which are shown in Fig. 2 (b). Using these primitives the characters are encoded into labeled trees. Fig. 2 (c) shows such an encoding for the characters shown in Fig. 2 (a). For ease of representation in the text, these characters shall be referred to by English characters which are phonetically closest to the original characters. For e.g. the Devanagari characters in Fig. 2(a) shall be referred to by the English characters G, H, and M respectively.

For our learning problem we have grouped a few handwritten characters into two classes. The positive class C^+ contains different handwritten versions of the same Devanagari character H. The negative class C^- could contain any handwritten character other than H. However to demonstrate the effectiveness of the algorithm, we have selected for C^-, Devanagari characters D, E and J which have a few features resembling those of the characters of class C^+. Fig. 3 (a) and (b) show the C^+ and C^- classes, respectively, constructed in such a manner. Fig. 4 shows the tree representation for the characters in C^+ and C^-(Fig. 3). Let us now proceed with the two learning processes, namely, concept learning and feature learning.



(a) Trees representing C+



(c) Trees representing C-

Fig. 4. Tree representation of C+ and C-.

6.2. Concept Learning

The objective of the concept learning phase is to discover the concept C in the positive class. The concept C could be a single labeled node or a collection of linked nodes (called macro) which helps in distinguishing the positive class. We start with the set of elementary operations and attempt to isolate the operation which gives the best class separation. For a good separation of the positive class we have to obtain a large $f_1(\omega)$ and a small $f_2(\omega)$. If none of the elementary operations give a good class separation, even after assigning the best weight distribution to these operations, then we proceed to build macro operations. This we do by generating all possible valid expansions of the elementary operation which gives the best class separation. By assigning a large weight to the potential macros we can check if there is any macro which give a better class separation. The macro(s) which we discover through this process of class separation would then give the class concept C.

While implementing such an optimization process, we observed that it is easier to build an algorithm which captures the macros by assigning a low weight to the macros. This is because a distance algorithm gives us the least cost path, and macros with low weight can only influence such a path. Therefore, we express the optimization process as,

$$\min_{\omega \in \Omega(t)} \{f_1(\omega) \mid f_{2p}(\omega) = k_1, \quad f_{2n}(\omega) = k_2\}$$

where $f_1(\omega)$ is the interclass distance,

$f_{2p}(\omega)$ is the average $\Delta_\omega(t)$ distance between the objects of the positive class at the cg of the simplex, which we shall term as the intra-positive class distance,

$f_{2n}(\omega)$ is the average $\Delta_\omega(t)$ distance between the objects of the negative class at the cg of the simplex, which we shall term as the intra-negative class distance, and

k_1, k_2 are positive constants.

Assuming that we have m and n trees in the positive and negative class respectively, the average intra-positive class distance $f_{2p}(\omega)$ and the average intra-negative class distance $f_{2n}(\omega)$ can be represented as:

$$f_{2p}(\omega) = \frac{2}{m(m-1)} \sum_{i=1}^{m-1} \sum_{j=i+1}^{m} \Delta_{\omega(t)}(T_i, T_j),$$

$$f_{2n}(\omega) = \frac{2}{n(n-1)} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \Delta_{\omega(t)}(T_i, T_j).$$

Also, the interclass distance $f_1(\omega)$ can be given as

$$f_1(\omega) = \min\{\Delta_{\omega(t)}(T_i, T_j) \forall 1 \le i \le m, 1 \le j \le n\}:$$

Iter.	Macro	$f_{1\min}(\omega)$	$f_{2n}(\omega)$
0	-	0.6667	0.4444
1	*dd*	0.3333	0.4444
2	*fdd*	0.1667	0.4444

Table 1. Results of Concept Learning

Application of the optimization algorithm to the example shown in Fig. 3, gives us $f_1(\omega)$ and $f_2(\omega)$ as shown in Table 1. In the first iteration, the value of the minimum $f_1(\omega)$ or $f_{1\min}(\omega)$ reduces from 0.666 to 0.333 with the discovery of the macro *dd*. Further expansion of the macro to *ddf* gives us the lowest value of $f_{1\min}(\omega)$ which equals 0.1667. The macro *ddf* represents the concept of the positive class. This can be verified by observing the trees of C+ and C- (see Fig. 4). The macro *ddf* (Fig. 5) is seen to be absent in C+ and present in all the trees of C+.

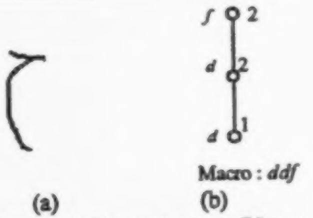


Fig. 5. (a) Concept macro (b)tree representation

6.3. Feature Learning

Discovery of Features F helps in subclassifying the positive class. The features are extracted using an optimization process similar to the concept extraction process. Since features F tend to make the positive class objects appear different from one another, assigning a low weight to the macros which represent F should reduce the intra positive class distance $f_{2p}(\omega)$. Therefore the feature learning optimization process can be represented as

$$\min_{\omega \in \Omega(t)} \{f_{2p}(\omega) \mid f_1(\omega) = k\},$$

where k is a positive constant.

For the example in Fig. 3, we observe three potential macro expansions *cc*, *cb* and *db* which do not occur in every character of C+. We compute $f_2(\omega)$ at the center of gravity of the simplex. For each potential macro, we compute $f_2(\omega)$ after assigning zero weight to that macro. The macro which gives the minimum value of $f_2(\omega)$ or $f_{2\min}(\omega)$ is considered as the best candidate for Feature F. In this example, we however notice that with either of the three macros the value of $f_{2\min}(\omega)$, fails to decrease. However, if we consider all the tree macros together and compute, the value of $f_2(\omega)$ is seen to decrease from 0.4444 to 0.333. This gives an indication that sufficiently powerful feature macros are not present in the positive class for the given example.

Iter.	Macro	$f_{2\min}(\omega)$	$f_1(\omega)$
0	-	0.4444	0.6667
1	*cc, cb, db*	0.3333	0.6667

Table 2. Results of Feature Learning

If we succeed in obtaining feature macros which reduce the value of $f_2(\omega)$ considerably, then these could be used for identifying the subclasses. In this handwritten character recognition example, if feature

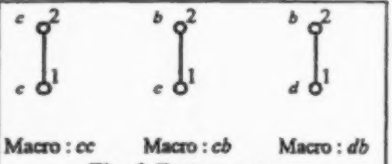


Fig. 6. Feature macros

F1 could be associated with handwriting Style1 and F2 with handwriting Style2, then a new handwritten character H of Style1 which has not been previously shown to the learning machine can be appropriately classified in two stages. In the first stage the new handwriting is compared using the concept C to check if it belongs to the positive class. Since a character H of Style1 would possess the concept C, we proceed to the next stage where, the features F1 and F2 are used to check for an appropriate match. Since the new character of Style1 would find a Feature F1 match, it can be appropriately classified. The values of $f_2(\omega)$ and $f_1(\omega)$ are used to indicate whether a successful match is obtained or not. This explains the recognition process.

7. Conclusions

This paper shows how the dynamic nature of ETTS makes it suitable for inductive generalization. The algorithm is shown to learn effectively with small training sets. We have shown how the distance function, which forms the basis for the ETTS, helps in extracting symbolic information from the training sets. With the help of an handwritten character recognition problem, we have demonstrated how the learning machine is capable of finding the characteristic concept C of the class C+, which enables it to distinguish members of this class from other classes. Storage of the positive concept rather than all the positive class examples provide considerable savings in storage space, and in fetch and retrieval time durations. We have also demonstrated use of the features F help in subclassification of the positive class.

The ETTS model, through the symbiosis of the symbolic and the vector space paradigm, gives a promising future to the development of powerful learning machines. The vector space approach provides the necessary robustness and noise tolerance while the syntactic approach provides the necessary descriptive power.

REFERENCES

[1] L. GOLDFARB, *What is distance and why do we need the metric model for pattern learning*, Pattern Recognition 25 (1992), pp. 431-38

[2] L. GOLDFARB, AND S. NIGAM, *The Unified Learning Paradigm: A Foundation for AI*, to appear as a chapter in a forthcoming book, eds. V. Honavar and L. Uhr, Academic Press, N.Y.

[3] O. GOTOH, *An inproved algorithm for matching biological sequences*, Journal of Molecular Biology. 162, (1982), pp.705-708.

[4] P. N. JOHNSON-LAIRD, *The Computer and the Mind: An introduction to cognitive science*, Harvard University Press, Cambridge, Massachusetts, 1988.

[5] V. N. KAMAT, V. C. BHAVSAR AND L. GOLDFARB, *Obtaining all occurrences of a tree pattern match*, in preparation for submission to Information Processing Letters, Holland.

[6] V. N. KAMAT, V. C. BHAVSAR AND L. GOLDFARB, *Generalized operation based Edit Distance for Ordered Trees*, in preparation for submission to SIAM J of Comp.

[7] J. B. KRUSKAL AND D. SANKOFF, *An anthrology of algorithms and concepts for sequence comparison*, Time Warps, String Edits, and Macromolecules: The theory and practice of sequence comparison. Editors Sankoff and Kruskal, Addison-Wesley, 1983.

[8] V. I. LEVENSHTEIN, *Binary codes Capable of Correcting Deletions, Insertions, and Reversals*, Cybernetics and Control Theory 10(8), 1966, pp.707-710; orig. Russian Doklady Academii Nauk SSR 163(4), 1965, pp. 845-848.

[9] G. A. MILLER, The magical number seven, plus or minus two: Some limits on our capacity to process information, Psychological Review, 63 (1956), pp.81-97.

[10] I. B. MUCHNIK, *Simulation of Process of Forming the Language for Description and Analysis of the Forms of Images*, Pattern Recognition, Vol. 4 (1972), pp. 101-140.

[11] W. B. SANTOSO, *Learning Algorithm for the ReconFig.urable Learning Machine*, Master's thesis, Faculty of Computer Science, Univ. of New Brunswick, Canada, 1992.

[12] S. WATANABE, *Pattern Recognition: Human and Mechanical*, Wiley, New York (1985).

[13] K. ZHANG, D. SHASHA, *Simple fast Algorithms for the Editing Distance between Trees and Related Problems*, SIAM J. Comput., 18 (1989), pp.1245-1262.

909743

General Principles in the Design of Real-Time Knowledge-Based Systems for the Threat Evaluation and Weapon Assignment Process

Robert L. Carling[1]
DREV[2], P.O. Box 8800,
Courcelette, Québec, G0A 1R0.

#149640

Abstract

The threat evaluation and weapon assignment (TEWA) process is an important process in the command and control system of an anti-air warfare (AAW) destroyer or frigate, since it ranks the air threats detected by the ship's automatic track management system (ATMS) according to specific criteria depending on the threat and the ship's characteristics and then assigns a hardkill or a softkill weapon to these threats in order to destroy or decoy them. A nonreal-time knowledge-based system has been built in SMALLTALK 80/HUMBLE to do threat evaluation and weapon assignment for a single stationary AAW destroyer attacked by anti-ship missiles. The knowledge-based system has been further refined to produce TEWA decisions for a manoeuverable ship attacked by anti-ship missiles.

This paper describes the algorithms of the TEWA knowledge-based system that integrate the use of hardkill/softkill weapons with the ship's movement functions.

This paper presents the real-time requirements of a knowledge-based system integrated into the air combat system of an AAW frigate. The high and low level designs of a real-time knowledge-based system for TEWA in terms respectively of partitioning artificial intelligence (AI) systems and assigning tasks are also presented. A transputer farm implementation of the threat evaluation function and a mapping of weapon assignment knowledge bases onto transputer networks are also described in qualitative terms. An alternative real-time implementation of AI tasks using Reduced Instruction Set Computer (RISC) processors and the associated problems of multiprocessor scheduling are also discussed.

Introduction

The design of the knowledge-based system for the TEWA function of a stationary AAW destroyer or frigate attacked by anti-ship missiles has been documented in references [1], [2] and [3]. The knowledge-based system comprises four knowledge bases called the Threat Evaluation knowledge base, the Result Evaluation knowledge base, the Force Resources Evaluation knowledge base and the Candidate Reaction Evaluation knowledge base. The first knowledge base comprises about fifty rules, while the remaining three knowledge bases together contain about fifty rules. The knowledge bases use backward chaining, multiple confidence factor paradigms, temporal reasoning and spatial reasoning. References [1] to [3], describe how threat ranking values are calculated from identity, kinematic, radar state, and engagement status parameters using the HUMBLE uncertainty calculus. They also describe how resource allocation plans are formed for hardkill and softkill weapons, how these plans are ranked and then how they are implemented against anti-ship missile threats. These knowledge bases and the plan formation and ranking will be known as the high level design of the knowledge-based TEWA.

A multiple ship anti-air warfare simulator is now being developed in SMALLTALK 80, in which each ship has a knowledge-based TEWA and the ships may manoeuvre before deploying hardkill or softkill weapons. The design of the anti-air warfare simulator is presented in reference [4].

Neither the initial knowledge-based system nor the multiple ship anti-air warfare simulator are real-time simulations. A resource loading study was undertaken to measure the effect of replacing the conventional TEWA in a TRUMPlike ship by artificial intelligence functions. The command and control computers, the AAW functions, the SHINPADS bus and the LINK 11 system are modelled in reference [5], while the artificial intelligence functions replacing the TRUMP TEWA are described in reference [6]. A Simscript model was devised to simulate anti-ship missile attacks on convoys of TRUMPlike ships. The simulation models the flow of messages from the sensors and weapons to the command and control computers and it also models the queue sizes of tasks arriving in the interface of each command and control computer. This model is described in reference [7]. Results from scenarios in which a small number of anti-ship missiles attacked three TRUMPlike ships and a very large number of anti-ship missiles attacked each ship of a four ship convoy have been obtained. In the latter scenario, there were very large input and output queues to the command and control computers of all ships in which AI functions had replaced the conventional TEWA. It, therefore, was concluded that faster processors and different architectures would be required in order to reduce the input and output queue sizes of command and control computers executing AI functions. The two possible candidates which could conceivably replace the current processors on board the TRUMPlike ship are transputers or RISC processors.

---

[1] Phone (418) 844-4432.

[2] DREV = Defence Research Establishment Valcartier.

TEWA Reactions for a Manoeuverable Ship

The AAW simulator is being developed for several warships which can manoeuvre when attacked by anti-ship missiles. Ship rotation will be used to support hardkill reactions. More specifically, ship rotation may occur in order to unmask a fire-control radar or naval gun that is in a blind zone.

In order to design a TEWA that supports rotation before hardkill deployment, the following steps will be taken. The reactions that are considered unengageable because there are blind zone problems will be noted. The amount of rotation required to make each reaction engageable is calculated. An attempt is made to create engageable reactions before considering hardkill reactions that involve rotation. A certain number of penalty points are associated with the plan when a component reaction involves a rotation because of the delay incurred before it can be implemented. Whenever, a plan P has a constituent reaction R that incorporates a rotation, calculations are made to determine whether other reactions in the same plan which are engageable would be made unengageable as a result of the rotation. Now, consider a plan P = [R1, R2, R3] where R1 is a reaction that involves a rotation while the others do not involve rotation. Further, assume that P has a score of X according to the existing methods of scoring plans. Suppose that it is determined that the implementation of R1 would make R2 unengageable. In such a case, the plan P is modified to be [R1, R3] and a new score is calculated including the previous penalty factor. Consider a plan P = [R1, R2, R3] where R1 and R3 both require rotation. It is always possible that plans may be drawn up (e.g., plan P) that have more than one reaction involving rotation. Clearly, both rotations cannot be implemented. In this case, the rotation is chosen which is associated with the reaction assigned to the highest ranked threat.

With the current TEWA design of the AAW simulator, precedence is given to the implementation of hardkill reactions including those needing rotation before the implementation of softkill reactions. The most general form of softkill reaction which is typical of chaff, jammers and passive decoys is a rotation of the ship to place it in the best position for softkill weapon deployment followed by a change in ship velocity to ensure feasibility of the softkill weapon deployment. After the weapon is deployed, another rotation and change in ship velocity give the ship the best chance to leave the threat-weapon engagement zone without any damage. In the last example given above, i.e., in the plan P consisting of [R1, R2, R3] where both R1 and R3 require rotations, R3 could be a hardkill reaction and R1 could be a softkill reaction. Thus, R3 is assigned to the most highly ranked threat and if the effect of implementing R3 does not prevent the softkill reaction R1 from being implemented, it will also be executed by the ship.

The Real-Time TEWA Problem

Many designers of successful real-time artificial intelligence systems have adopted a hierarchical approach. The designers of the EAVE (Experimental Autonomous VEhicle) robot at the National Research Council adopted a real-time AI model in which a rule-based system provided the high-level decision making and decided in real-time on which actions the robot platform should take. The details of this design and platform task levels are given in references [8] and [9]. Reference [10] also mentions a systematic way of designing knowledge-based systems at two different levels. In this article, it suggests that many AI problems are really complex generic tasks, i.e., they are further decomposable into components that are more elementary in the sense that each of them has a homogeneous control regime and knowledge structure. The high level design of our knowledge-based system has been explained in the introduction. The three low level implementations of the TEWA that could conceivably execute knowledge bases in real-time are a single fast processor, transputer networks or distributed RISC processors.

The knowledge bases of the TEWA will have to interact with various combat system elements. In an air engagement, the TEWA will receive data from various sensors and LINK 11. The data coming from these sensors and LINK 11 can be modelled by synchronous or asynchronous processes and must be processed by the TEWA knowledge bases within a certain real-time constraint. As soon as the threats have been identified to be hostile, the TEWA knowledge bases must do engageability calculations within a certain real-time requirement and then send the engagement orders to the designated weapon. If softkill weapons are going to be used, then the TEWA must send a request for information to the EWCP (Electronic Warfare Control Processor) which replies to this request according to the real-time design of the

EWCP. There will be a real-time requirement for the total time required to send a request from TEWA to the EWCP, to have it processed by the EWCP and send an answer back to TEWA. In preparation for missile launch or gun firing and after missile or gun intercept with the target, there will be many messages sent from TEWA to weapon system elements and vice-versa. Each of these must be executed within a certain real-time constraint.

Up to the present moment, the author has not been able to find any document specifying upper bounds for the real-time requirements described in the previous paragraph. The only documents available are those giving the global reaction time of the ship from fire-control radar designation to surface-to-air missile (SAM) launch when the ship is faced with a quick reaction threat. In references [5] and [6], Thomson CSF has made estimates for the TEWA, sensor and weapon element processing times when messages are created, are relayed or terminate at these combat system nodes. These processing requirements were devised for moderate threat loads and for a TRUMPlike ship equipped with militarized processors. References [5], [6] could be used as a starting point for the TEWA real-time requirements, but these values may have to be decreased for heavier threat loads.

Loading Knowledge Bases onto Transputer Networks

The real-time implementation of the high-level TEWA design can be done so that some of the knowledge bases are executed concurrently and some are executed sequentially. An implementation can be attempted where the Threat Evaluation, Result Evaluation and Force Resources Evaluation knowledge bases are evaluated concurrently and then the Engageability, Effectiveness and Reaction Ranking of resource allocation plans are calculated sequentially after the three initial concurrent computations. By the very nature of engageability calculations, the two subprocesses of Candidate Reaction Evaluation (i.e., engageability and effectiveness) that generate resource allocation plans and the Reaction Ranking of these plans have to be sequential. The other way of implementing the TEWA process is to execute sequentially each of the knowledge bases Threat Evaluation, Result Evaluation, Force Resources Evaluation, Candidate Reaction Evaluation and the Reaction Ranking function. It is to be noted that although the Candidate Reaction Evaluation and Reaction Ranking processes are sequential, there is some inherent parallelism in them. When engageability and effectiveness are being calculated for hardkill and softkill weapons, they can be done concurrently, since they are completely independent of each other. At the present moment, with the current way that plan ranking takes place, it is feasible to do simultaneous reaction ranking by using duplicate reaction sets for a ship that does not rotate and for a ship that needs to rotate before deploying hardkill or softkill weapons. Since transputers may be used for concurrent and pipelined applications, some general steps will be presented which will have to be performed before any transputer implementation of knowledge bases can take place.

In order to make the Threat Evaluation process as rapid as possible for a large number N of tracks, the ideal situation is to try and make the time for calculating the threat levels for N tracks be equal to the time for calculating the threat level of one track since the N tracks in our current threat evaluation model are basically independent of each other. The Threat Evaluation problem has the characteristic that the same set of instructions (namely, the fifty rules of the knowledge base) are applied to each track. This kind of problem is known in the parallel processing literature as the "farming" problem and according to reference [11] is particularly well adapted for implementation in a network of transputers.

The Candidate Reaction Evaluation and Reaction Ranking processes are sequential with some inherent parallelism. Unlike the Threat Evaluation process, these two processes are not farming processes in the sense that the same set of instructions is applied to the incoming data. If an incoming track is not engageable, it is immediately discarded from the pipelined processes of Engageability, Effectiveness and Reaction Ranking. In the case of the Result Evaluation and Force Resources Evaluation knowledge bases, updates are sent to them each time the result of a reaction or the status of a resource are known and in this case only the pertinent rule is fired. Thus, these two knowledge bases also cannot be considered to be "farming" processes.

Reference [11] recommends the following steps for determining a transputer network which will satisfy the real-time requirements of the AAW problem. These are:

Step 1 : Functional decomposition
Step 2 : Partitioning and Allocation
Step 3 : Scaling.

These three steps could well be applied to the Threat Evaluation, Result Evaluation, Force Resources Evaluation and the Candidate Reaction Evaluation /Reaction Ranking knowledge bases. In the functional decomposition stage, it is necessary to identify the set of logical modules and data files making up a transputer implementation of the problem. If the Threat Evaluation problem is considered as a "farming" process, the master process must be identified along with the worker processes. One possible formulation is to model the master process as being the fused track database and the worker processes as separate copies of the Threat Evaluation knowledge base.

In the partitioning stage, the logical modules and data files that are the output of the functional decomposition are partitioned, i.e., are mapped into a set of parallel processes executing on their own data and on non local data exchanged with other processes by means of a suitable communication network. Two objectives now have to be met. First, the grain size of the partitioning into parallel processes (defined to be the amount of computation assigned to each process) must be maximized in order to obtain a short computation time for the relevant process. Secondly, if the grain size is too small, there will be large delays caused by interprocessor communication which will cause the total execution time to be large. Thus, a grain size has to be obtained which is small enough to keep the computation time fast but not so small that the communication times are too large. In the case of the Candidate Reaction Evaluation knowledge base, the fastest solution can often correspond to a partition with a physical meaning. In this case, there is a natural partitioning between reactions that are entirely hardkill, that are entirely softkill, involve rotation before hardkill/softkill deployment or do not involve rotation before hardkill/softkill deployment. In addition to determining the grain size of partitioning, the transputer topology must be chosen. The transputer topology is often chosen as a function of the physical problem . Many choices have been put forward such as ring topologies, square mesh topologies, and hypercubes (see reference [11]). However, references [12] and [13] suggest pipelined architectures of transputers for pipelined problems and any network (small or large) of transputers in which members are connected to each other by at most four communication channels for a general problem. The transputer network is usually implemented using a host computer and a root transputer, i.e., the tasks from the ship combat system are sent to the host computer and then to the root transputer that acts as a system manager routing each task to the appropriate transputer for execution. The topology that is chosen will obviously influence the kind of routing algorithm used for each task. Routing systems for transputer farms are explained in reference [13], p.205, while message passing from the root transputer to general transputer networks are given in reference [12], pp. 139-156. In general, the topology chosen should be compatible with the grain size, and the routing scheme chosen should be the most efficient for the given topology.

In general, according to reference [11], the problem of software allocation of n processes to m processors is NP complete. In this particular case, there may be more than one solution. The most important point is to find a solution that satisfies the real-time requirements for Threat Evaluation, for Result Evaluation, for Force Resources Evaluation and for Candidate Reaction Evaluation/Reaction Ranking and to prove that this solution satisfies the above real-time requirements.

After going through the process partitioning stage and the software to processor allocation stage, a design for a topology of transputers with an adequate communication system and with knowledge bases assigned to the transputers will have emerged so that the total execution time for each knowledge base is minimized for a large number N of tracks. However, it may still be slower than the real-time requirement. If this is the case, a scalability study will be undertaken. The goal is to find the number of processors that makes it possible to obtain a performance that is as close as possible to the desired one without making the system inefficient. The speed-up obtained by using a parallel architecture is not a linear function of the number of processors used: as the number of processors increases, the diameter of the network interconnecting them and the distances between communicating processing elements also grow. Furthermore, as the size of the problem is usually fixed, there is an inevitable reduction in the grain size. This means that, starting from a given system configuration, performance improvements due to the addition of a new processor become smaller and smaller, efficiency decreases and the cost/performance ratio grows.

In practice, for scaling to be performed correctly, it is important to determine the number of processors representing the threshold beyond which the speed-up is no longer linear and the computing resources are used inefficiently. In real-time systems, the objective is to obtain a given performance and it may, therefore, be reasonable to choose a high number of processors, even if used inefficiently. During the scalability study, it is also necessary to do a performance analysis of the transputer system. At present, the most widely used indices are speed-up and efficiency. The speed-up obtained for a certain problem by using a parallel computer is commonly defined as the ratio between the response time for that problem using a single processor and the response time when a parallel computer made up of K processors is used. If the latter of these times is indicated by $T_{par}(K)$ and the one measured under completely sequential conditions by T_{seq}, then obviously $T_{seq} = T_{par}(1)$. Using this notation, the speed-up S (which is, of course, a function of K) is given by the expression:

$$S(K) = T_{seq}/T_{par}(K)$$

Efficiency ε is usually used along with speed-up as a performance index, and is defined as follows:

$$\varepsilon = S(K)/K$$

This definition of efficiency presupposes that processors are the critical resource of the system. It is the most accepted definition of efficiency and does not take into consideration any of the other important factors that contribute to efficiency such as : algorithmic overhead, software overhead, uneven load balancing, communication overhead and external communication overhead.

In order to study the above proposed mappings and their interaction with interrupts coming from the ship's combat system, a Petri-Net simulation can be used as in reference [14]. The results of some parallel processing studies show that good scalability can be obtained for certain architectures, transputer types and host computers but when these are changed the scalability is not so good (see reference [15]).

RISC Processor Implementations of Artificial Intelligence

The information flow from the ship's sensors, weapons and LINK 11 to the AI TEWA functions during an anti-ship missile attack can be modelled by periodic processes. However, some sensors such as ESM only update their state and hence send an interrupt to TEWA each time the external environment changes. In existing real-time fusion systems, the current practice is to associate the incoming asynchronous data with the appropriate cycle of periodic data (such as radar). As soon as the anti-ship missile threats become engageable by the hardkill and softkill weapons, the number of tasks received by the AI TEWA functions on the ship's processors increases. Suppose that the processors of the TRUMPlike ship are replaced by hypothetical RISC processors. One of the very important considerations determining how many functions will be loaded onto a RISC processor, and which ones will be loaded, is the scheduling of these functions on the multiprocessor architecture for the command and control system of the ship. If we assume that data coming from all sensor, weapon and Link sources is periodic, important results developed for the rate monotonic scheduling of uniprocessors made be used to schedule AI functions on multiprocessors.

The first important result concerning the rate monotonic scheduling of a single processor was discovered by Liu and Layland and published in reference [16]. Their theorem states:

Theorem(Liu and Layland)

A set of n independent periodic tasks scheduled by the rate monotonic algorithm will always meet its deadlines, for all task phasings, if

$$(C_1/T_1)+ \dots +(C_n/T_n) \leq n(2^{1/n}-1) = U(n) \qquad (1)$$

where C_i and T_i are the execution time and period of task τ_i.

The above theorem offers a sufficient condition that characterizes schedulability of a task set under the rate monotonic algorithm. This bound converges to 69 percent (ln 2) as the number of tasks approaches infinity. The values of the scheduling bounds for one to ten independent tasks are as follows:

$U(1) = 1.0$, $U(2) = 0.828$, $U(3) = 0.779$
$U(4) = 0.756$, $U(5) = 0.743$, $U(6) = 0.734$
$U(7) = 0.728$, $U(8) = 0.724$, $U(9) = 0.720$
$U(10) = 0.7177$

References [17] and [18] explain how to use this theorem for simple examples and how the assumption on periodicity can be relaxed by the introduction of sporadic servers for aperiodic tasks. The algorithms for sporadic servers can be used to program interrupts for multiprocessor systems (see reference [19]). The condition of the Liu and Layland theorem is not however necessary as shown in reference [20].

Multiprocessor Scheduling

The basic idea of this section is to try and extend some of the scheduling results from a single processor to several processors in order to develop a static scheduling theory for multiprocessors. Rate monotonic algorithms are better suited to static scheduling of multiprocessors (each processor runs the same set of tasks throughout an air engagement) than to dynamic scheduling of multiprocessors (see reference [21]). The following theorem will be stated and some indications will be given of its proof:

Theorem 1

If $\{e_i\}$, $i=1,\dots,k$ is a set of k periodic tasks and $\{E_i\}$ is a set of processors each having the same processing speed and for which the execution time C_i of a task is less than or equal to the arrival time T_i, then there always exists a minimal number m of these processors such that for each processor of the set $\{E_i\}$, $i=1,\dots,m$ inequality (1) is true for all the tasks assigned to E_i.

Proof : The theorem may be proved by ordering the utilizations C_i monotonically, assigning the tasks to each one of the processors so that the Liu and Layland conditions are satisfied and then starting from the tasks assigned to the first processor systematically rearranging them among the remaining processors. Each time a rearrangement is successful the number of processors decreases by one. Eventually, the process of reducing the number of processors executing all tasks will stop and this will give the minimal number m.

In many cases, T_i is smaller than the execution time C_i. In order to answer scheduling problems where $C_i > T_i$, the following theorem can be proved.

Theorem 2

If there are k tasks $\{e_i\}$ satisfying $T_i < C_i < 2T_i$, then m = 2k will give a set of processors that can schedule all tasks but k of these processors will be inactive for $n(2T_i - C_i)$ ms where n = ceiling($T/2T_i$) for some finite time interval T. Moreover, if the inactive time of the above m processors are used to schedule tasks satisfying $C_i > 2T_i$ then the deadline for completing these tasks occurs after $4T_i$.

Proof

If D_i is the deadline for completing task e_i, the fact that

$$D_i > (2T_i/T_j)2T_j = 4T_i$$

follows from

$$n(2T_j - C_j)/C_i > 1.$$

In view of theorem 2, it is better to execute tasks of long duration, i.e., tasks for which $C_i \geq 2T_i$ on a faster processor in order to satisfy its deadline. The next question that arises is how fast should this processor be. The following theorem will provide an answer to this question.

Theorem 3

Let $\{e_i\}_{i = 1,\dots,k}$ be a set of k tasks with execution times C_i and arrival times T_i on a set of processors E_i. Define

$$N = \max\left(5, \max_{i = 1,\dots,k} \text{ceiling}(C_i/T_i)\right) \qquad (2)$$

If F_i is a set of processors that can reduce the execution times of the tasks e_i to C_i/N, then

(1) there exists a minimal set of processors of type F_i for which the Liu and Layland conditions are satisfied,

(2) the number of processors of type F_i is considerably less than the number used in theorem 2,

(3) all tasks for which $C_i > 2T_i$ will satisfy their deadlines, i.e., $C_i/N \leq T_i$.

Proof

The three above statements can be deduced from equation (2).

Example:

The following data was taken from a simulation where several anti-ship missiles attacked a TRUMPlike ship. The execution times C_i and arrival times T_i of tasks for all AAW functions are given in milliseconds. The TEWA design, i.e., the distribution of AAW functions to processors was chosen arbitrarily and is not necessarily ideal.

AAW function	C_i (msec)	T_i (msec)
CPCI1		
Navigation I/F	1.00	14.0845
Rapidly Firing Gun I/F	4.62	28.169
Fire-Cntrl Radar I/F	11.20	9.337
GunFCR I/F	11.20	15.29
Target Data	7.89	3.3647
CPCI2		
ATMS I/F	3.60	29.112
Target Evaluation	41.35	69.444
CPCI3		
WDS I/F	9.37	6.4226
Result Evaluation	22.50	15.987
Candidate Reaction Evaluation	25.37	15.72327
Ranking of Candidate Reactions	6.96	25.00
Reaction Selection	3.09	10.288
CPCI4		
LINK 11 I/F	9.46	5.68182
CPCI6		
ESM I/F	40.20	9.19118
CHAFF I/F	1.01	30.30303
Force Data	1.92	2.66064
Forces Resources Evaluation	7.50	14.9365198

The utilizations of the seven functions for which the execution time C_i was greater than the arrival time

T_i were calculated in the ship data problem and are given below.

AAW Function	Utilization($x_i = C_i/T_i$)
Fire-Cntrl Radar I/F	1.199529
Target Data	2.344934
Weapon Direction System I/F	1.4589107
Result Evaluation	1.4073935
Candidate Reaction Evaluation	1.613532
LINK 11 I/F	1.6649595
Electronic Support Measures I/F	4.373758

When theorem 3 is applied to the seventeen AAW functions of the example, and each of their task execution times is divided by 5, the following assignment of tasks to the processors $\{F_i\}$ is obtained.

F_1 : $.006666<.0142<.024732<.032802<.05568$
$<.06007<.100425<.119088876<.1443262$
$<.146501$

$$\sum_{i=1}^{10} x_i = 0.704491 < U(10) = .7177$$

F_2: $.239906<.2814787$

$$\sum_{i=11}^{12} x_i = .5213847 < U(2) = .828$$

F_3: $.29178214<.3227064$

$$\sum_{i=13}^{14} x_i = .61448854 < U(2) = .828$$

F_4: $.3329919<.4689868$

$$\sum_{i=15}^{16} x_i = .80197872 < U(2) = .828$$

F_5:. $.8747516641$

$$\sum_{i=17}^{17} x_i = .8747516641 < U(1) = 1$$

Finally, an example will be given of a data set where rearrangement in theorem 1 is actually necessary. If the artificial intelligence functions of the example are considered separately and their utilizations x_i are calculated, the following assignment to three processors $\{F_i\}$ is obtained.

F_1: $.05568<.06007<.100425<.119088876$
$<.1443262$

$$\sum_{i=1}^{5} x_i = .479590076 < U(5) = .743$$

F_2: $.2814787< .3227064$ (3)

$$\sum_{i=6}^{7} x_i = .6041851 < U(2) = .828$$

F_3: $.4689868$

$$\sum_{i=8}^{8} x_i = .4689868 < U(1)$$

The AI functions considered here are : Ranking of Candidate Reactions, Reaction Selection, Force Resources Evaluation, Target Evaluation, Force Data, Result Evaluation, Candidate Reaction Evaluation, Target Data. The utilizations obtained firstly by truncating those of the system (3) to two decimals, except in the case of .119088876 which is rounded to .12 and secondly by changing the utilization on F_3 to .36 instead of .4689868 can be shown to correspond to a task set that can be rearranged on a smaller number of processors. The original assignment as defined in theorem 1 gives:

F_1: $.05<.06<.1<.12<.14$

$$\sum_{i=1}^{5} x_i = .47 < U(5) = .743$$

F_2: $.28 < .32$

$$\sum_{i=6}^{7} x_i = .6 < .828$$

F_3: $.36$

$$\sum_{i=8}^{8} x_i = .36< 1$$

These tasks may be rearranged on F_2 and F_3 thus:

F_2: $.05<.1<.28<.32$

$$\sum_{i=1}^{4} x_i = .75 < U(4) = .756$$

F_3: $.06<.12<.14<.36$

$$\sum_{i=5}^{8} x_i = .68 < U(4) = .756$$

Thus, the eight tasks have been rearranged on two processors.

CONCLUSIONS

In this paper, algorithms are put forward to describe the hardkill and softkill reactions that can be chosen by the knowledge-based TEWA of a manoeuverable ship which is attacked by anti-ship missiles. A methodology is also described for designing real-time knowledge bases on transputer networks. The description includes a discussion of some of the different viewpoints and difficulties of concurrent computing using transputers. Finally, some theoretical results are obtained for multiprocessor scheduling of artificial intelligence functions on RISC processors and these results are illustrated using some simple shipboard anti-air warfare AI functions.

References

1. Carling, R.L., "A Knowledge-Base System for the Threat Evaluation and Weapon Assignment Process", *Naval Engineers Journal*, vol. 105, no. 1, pp. 31-41, January 1993.
2. Carling, R., "An Artificial Intelligence Approach to the Threat Evaluation and Weapon Assignment Problem", *Maritime Engineering Journal*, pp. 12-16, January 1993.
3. Thomson CSF Systems Canada, "TEWA Simulator Design Report", DSS Contract No.: W7701-8-4770/01-XSK, TCSC Report No.: C0048C-01, July 1991.
4. Thomson CSF Systems Canada, "Software Design Document for the Modelled Entities of the Anti-Air Warfare Simulator", DSS Contract No.: W7701-0-3202/01-XSK, TCSC Report No.: G0456C-06, November 1992.
5. Thomson CSF Systems Canada, "TEWA Resource Loading Study, Ship AAW System Model Specification", DSS Contract No.: W7701-9-4137/01-XSK, January 1991.
6. Thomson CSF Systems Canada, "TEWA Resource Loading Study, Ship Convoy Model Specification", DSS Contract No.: W7701-9-4137/01-XSK, TCSC File No.: G0194C, August 1991.
7. Thomson CSF Systems Canada, "TEWA Resource Loading Study, Simscript Model Design Document", DSS Contract No.: W7701-9-4137/01-XSK, TCSC File No.: G0194C, January 1993.
8. Green, D., Liscano, R., and Wein, M.,"Real-Time Control of an Autonomous Mobile Robot using the Harmony Operating System", *Proceedings of the IEEE International Symposium on Intelligent Control 1989*, Albany, New York, pp. 374-378, 25-26 September 1989.
9. Liscano, R., Fayek, R.E., Karam, G.M., "A Blackboard, Activity-Based Control Architecture for a Mobile Platform", NRC Report No. 33196.
10. Chandrasekaran, B., "Generic Tasks in Knowledge Based Reasoning: High-Level Building Blocks for Expert System Design", *IEEE Expert*, pp. 23-30, Fall 1986.
11. de Carlini, U., and Villano,U., "Transputers and Parallel Architectures: message-passing distributed systems", Ellis Horwood Limited, Chichester, England, 1991.
12. INMOS, "The Transputer Applications Notebook: Systems and Performance", first edition, Bristol, England, June 1989.
13. INMOS, "The Transputer Applications Notebook: Architecture and Software", first edition, Bristol, England, May 1989.
14. Finn, A., Griffin, M., and McClurg,C., "Modelling and Simulation of an i860-Based Multiprocessor", *24th Annual Simulation Symposium*, New Orleans, LA, 1991.
15. Lina, J-M., Scott, C.K., Goulard, B., and Mayrand M., "Advanced Simulation of CANDU reactors at low cost with parallel processing", *17th CNS Simulation Symposium*, Kingston, 1992.
16. Stankovic, J.A., and Ramamritham, K., "Hard Real-Time Systems", IEEE Computer Society Press, Los Alamitos, CA, pp. 174-189, 1988.
17. Sha, L., and Goodenough, J.B., "Real-Time Scheduling Theory and Ada", *IEEE Computer*, pp. 53-62, April 1990.
18. Sprunt, B., Sha, L., and Lehoczky, J.P., "Aperiodic Task Scheduling for Hard-Real-Time Systems", *J. of Real-Time Systems*, Vol. 1, pp. 27-60, 1989.
19. Encore 93 Series Technical Summary, Encore Computer Corp., Fort Lauderdale, FL, 1991.
20. "The Handbook of Real-Time Systems Analysis: Based on the Principles of Rate Monotonic Analysis", *Carnegie Mellon University, Software Engineering Institute*, July 1992.
21. Rajkumar, R., Sha, L., and Lehoczky, J.P., "Real-Time Synchronization Protocols for Multiprocessors", *Proc. IEEE Real-Time Systems Symp.*, Los Alamitos, CA, pp. 259-269, 1988.

104945

Les stratégies de résolution de conflits utilisées avec une restriction de type temporel

Micheline Bélanger[1]
Centre de recherche pour la défense, Valcartier
2459, boul. Pie-XI nord (C.P. 8800)
Courcelette (Québec)
G0A 1R0

#149643

Résumé

Les systèmes experts basés sur les règles de production répètent itérativement trois étapes principales lors de leur exécution. Ces trois étapes sont: l'appariement de la mémoire de production avec la mémoire de données, la sélection de la règle à exécuter et l'exécution de la règle choisie. Le travail décrit ici se limite à la phase de sélection de règles réalisée par l'utilisation de stratégies de résolution de conflits.

Le but de ce travail est de cerner la ou les techniques de résolution de conflits qui est(sont) la(les) plus souhaitable(s) dans les applications comportant une restriction de type temporel. Plusieurs des stratégies utilisées habituellement dans les systèmes experts (e.g. le degré de priorité associé aux règles, le degré de complexité associé aux règles, l'âge des faits contenus dans les règles de l'ensemble de conflits, etc) ont été étudiées dans un contexte où les résultats doivent être obtenus à l'intérieur d'un temps t, déterminé à l'avance. Les avantages et les inconvénients de chacune des stratégies ont été répertoriés afin de prendre conscience des possibilités de chacune.

Les résultats obtenus de ces diverses études permettent de définir une stratégie de résolution de conflits propice aux applications ayant une contrainte temporelle à respecter. Par la suite, une combinaison de techniques de résolution de conflits a également été étudiées toujours dans un contexte comportant une contrainte de type temporel.

Introduction

Plusieurs systèmes experts sont basés sur les règles de production. Ce genre de système est formé d'un ensemble de règles localisé dans la mémoire de production ainsi que d'un ensemble de faits localisé dans une base de données appelée la mémoire de données. Le moteur d'inférence du système expert exécute en cycle trois opérations sur ces deux mémoires (voir la figure 1). Premièrement, il vérifie quelles sont les règles de la mémoire de production dont les conditions sont satisfaites par les faits contenus dans la mémoire de données. L'appariement de la mémoire de production avec la mémoire de données permet l'obtention d'un ensemble de règles appelé l'ensemble de conflits. Deuxièmement, le

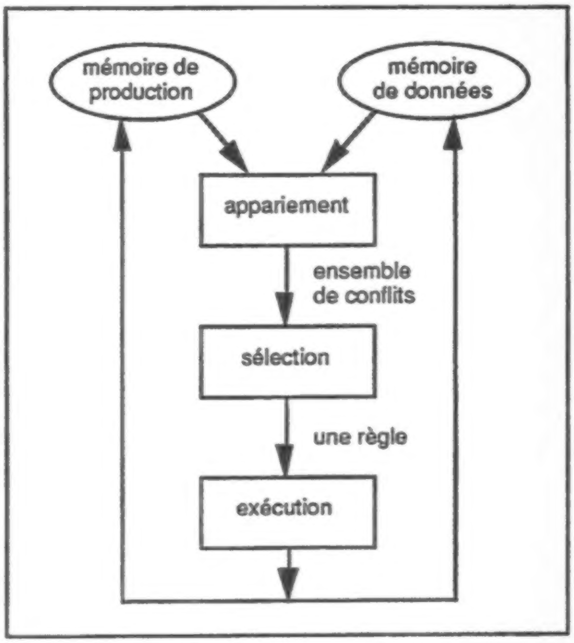


FIGURE 1: cycle d'exécution

---

[1] Téléphone: (418) 844-4379
e-mail: belanger@jupiter.drev.dnd.ca

moteur d'inférence fait la sélection de la règle à exécuter parmi les règles de cet ensemble en utilisant des techniques appelées stratégies de résolution de conflits. Finalement, il exécute la règle choisie. L'exécution d'une règle peut entrainer des modifications de la mémoire de production aussi bien que de la mémoire de données. Ces trois opérations sont donc réitérées aussi longtemps qu'il y a des règles actives dans l'ensemble de conflits.

Le lien entre la mémoire de production et la mémoire de données peut être représenté par un arbre de connaissances appelé l'espace de recherche du système expert. La figure 2 présente un arbre de recherche contenant les faits A et B ainsi que les trois règles:

R1: SI A ALORS C
R2: SI B ALORS E
R3: SI B ET C ALORS F

Les noeuds de l'arbre représentent les faits courants alors que les branches représentent les règles applicables. Les règles peuvent être exécutées selon différentes séquences. Ainsi, si le but du système expert de la figure 2 est de savoir si le fait F existe, il est plus rapide d'utiliser le parcours R1-R3 que les parcours R2-R1-R3 ou R1-R2-R3.

Dans plusieurs systèmes à caractère militaire, le temps de réponse est un des facteurs critiques de succès. Dans un tel contexte, les systèmes sont soumis à une contrainte de type temporel où le temps d'obtention de certaines réponses est limité. La restriction de type temporel est une des caractéristiques des systèmes experts en temps réel [Référence 1].

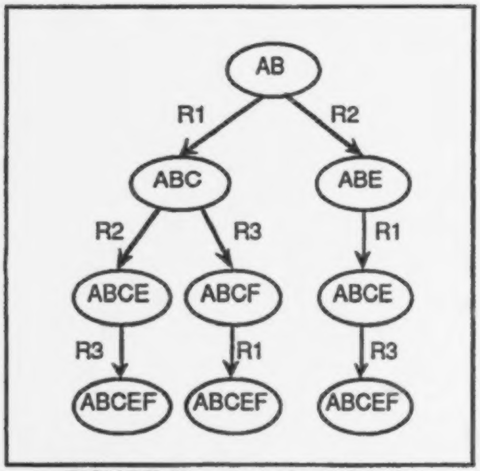


FIGURE 2: espace de recherche

Lorsqu'il est question de diminuer le temps de réponse d'un système expert, trois avenues sont habituellement considérées:

1) la réduction du temps nécessaire pour l'appariement des règles;
2) la réduction du temps nécessaire pour la sélection de la règle à exécuter; ou
3) la réduction du temps nécessaire pour l'exécution d'une règle.

La réduction du temps de sélection de la règle à exécuter s'avère un choix discutable dans le contexte qui nous intéresse. En utilisant une méthode de sélection plus rapide, il existe un risque que le choix ainsi fait soit moins approprié pour le problème à traiter en priorité. De plus, si le système expert doit atteindre simultanément plusieurs buts, la règle choisie peut ne pas être directement impliquée dans le processus de résolution qui nécessite une réponse à l'intérieur d'un temps limité.

Une façon d'éviter ce problème consiste à améliorer la phase de sélection de la règle à exécuter. Si, en ayant une meilleure phase de sélection, le moteur d'inférence réduit le nombre de cycle du système expert avant l'obtention de la solution désirée (parcours plus rapide), il devrait en résulter une réduction de temps de réponse. Il faut donc être en mesure d'éviter que le système expert perde un temps précieux à exécuter des raisonnements qui ne sont pas absolument nécessaire à l'obtention des réponses devant respecter des contraintes de temps. Dans cette optique, il devient acceptable de prendre plus de temps pour choisir une règle, puisque cela implique de ne pas faire de raisonnements qui pourraient être exécutés en second lieu.

L'idée de base consiste donc à exécuter en premier les règles impliquées dans le processus d'obtention de réponse le plus restreint dans le temps. Par la suite, le moteur d'inférence peut exécuter les autres règles de l'ensemble de conflits dont les contraintes de temps sont plus larges.

Les stratégies de résolution de conflits

Une des fonctions du moteur d'inférence d'un système expert est de choisir dans l'ensemble de conflits la prochaine règle qui sera exécutée. Pour ce faire, le moteur d'inférence utilise une stratégie de résolution de conflits. Il existe plusieurs de ces stratégies. Cette section décrit les stratégies de résolution de conflits que l'on retrouve habituellement dans les systèmes experts [Référence 2].

Le degré de priorité associé aux règles

Dans cette stratégie de résolution de conflits, le programmeur associe un degré de priorité à chacune des règles de la base de connaissances. Lors de la phase de sélection, le moteur d'inférence choisit, parmi toutes les règles de l'ensemble de conflits, celle qui a le degré de priorité le plus élevé. Par exemple, les degrés de priorité peuvent être représentés par des entiers échelonnés de -100 à +100, où la valeur +100 est la priorité la plus élevée et -100 la moins élevée.

Cette stratégie est, sans contredit, très importante puisqu'elle permet au programmeur d'indiquer un ordre de priorité d'exécution des règles. Grâce à cette stratégie, le moteur d'inférence choisit les règles qui sont les plus importantes à exécuter en premier. Cette stratégie met l'emphase sur les règles et non pas sur les faits contenus dans les règles. Il est néanmoins possible au programmeur de donner un degré de priorité en fonction du type de faits contenus dans la règle. Par contre, lorsque la même règle est activée plus d'une fois mais avec des faits différents, ou lorsque plus d'une règle a le même degré de priorité, cette stratégie ne permet pas de désigner la règle qui devrait être exécutée en premier.

Le degré de complexité associé aux règles

Le degré de complexité associé aux règles représente le nombre de clauses du côté gauche de la règle. Pour certains moteurs d'inférence de systèmes experts, il peut également représenter le nombre de prédicats de chacune des clauses. Il y a deux techniques reliées au degré de complexité associé aux règles: soit le moteur d'inférence choisira, parmi les règles de l'ensemble de conflits, la règle qui possède le plus haut degré de complexité; soit il choisira la règle qui possède le plus bas degré de complexité.

La stratégie du degré de complexité associé aux règles exploite l'idée que plus une règle a de clauses, plus elle est spécifique, ou vice versa. Cette stratégie n'apporte rien d'avantageux puisque leur fondement est tout-à-fait arbitraire. Le nombre de clauses que possède une règle ne donne aucune indication contenant l'ordre dans lequel devraient être exécutées les règles ou devraient être traiter les faits. L'utilisation de cette stratégie apparaît donc sans valeur significative.

L'âge des règles de l'ensemble de conflits

L'âge d'une règle de l'ensemble de conflits correspond au temps écoulé depuis que la règle a été placée dans l'ensemble de conflits jusqu'au moment présent. Il y a deux différentes techniques reliées à l'âge des règles de l'ensemble de conflits. Soit le moteur d'inférence, lors de la phase de sélection, choisit la plus vieille règle de l'ensemble de conflits, soit il choisit la plus jeune règle.

Dans le premier cas, on considère que plus une règle est vieille, plus il est important de l'exécuter en premier. Dans le second cas, l'idée qu'une règle activée récemment est plus importante qu'une règle activée plus tôt est exploitée. L'idée peut être intéressante dans certaines situations où il est toujours plus importants de travailler sur les raisonnements les plus récents.

Le problème avec ces stratégies est qu'il n'y a aucune référence sur l'importance devant être accordée à la nature de la règle même ainsi qu'aux faits qu'elle contient.

L'âge des faits contenus dans les règles de l'ensemble de conflits

Dans cette stratégie, le moteur d'inférence tient compte de l'âge des faits contenus dans les règles de l'ensemble de conflits. L'âge des faits est calculé à partir de leur création. Il y a donc essentiellement deux techniques reliées à l'âge de faits contenus dans les règles de l'ensemble de conflits. Le moteur d'inférence, lors de la phase de sélection, peut choisir la règle de l'ensemble de conflits contenant les plus vieux faits. Cette stratégie implique que plus un fait est vieux plus il est important. Le moteur d'inférence peut également choisir la règle de l'ensemble de conflits contenant les faits les plus jeunes. Cette dernière stratégie implique que les faits les plus récents sont plus importants que les plus vieux.

L'utilisation de l'âge des faits contenus dans les règles permet de diriger le raisonnement du système expert en fonction de l'importance qu'ont les faits entre eux. Par contre, il n'est nullement question de l'importance que peuvent avoir les règles entre elles.

Le hasard

Dans cette stratégie, on laisse le hasard choisir la prochaine règle qui sera exécutée. Parfois, le meilleur choix est fait, parfois le pire. Il n'y a aucune référence à l'importance rattachée aux règles ou aux faits.

La première règle

Dans cette stratégie, le moteur d'inférence choisit la première règle de l'ensemble de conflits. Cette stratégie est celle qui prend le moins de temps à s'exécuter. Elle peut être considérée équivalente au hasard puisqu'il n'y a aucune notion de direction dans le choix des règles.

Les combinaisons de stratégies

Il arrive fréquement qu'une seule stratégie de résolution de conflits ne soit pas suffisante, i.e. que la stratégie utilisée sélectionne plus d'une règle. Par exemple, plus d'une règle peuvent avoir le même degré de priorité ou le même âge des faits. Dans ces circonstances, il faut appliquer une seconde stratégie de résolution de conflits sur ce sous-ensemble de règles. Il est même possible qu'une troisième stratégie soit nécessaire.

La seconde et la troisième stratégies doivent, tout comme la première, être choisies en fonction des besoins de l'application.

Application type

L'application qui nous intéresse concerne l'identification des pistes (traduction libre de "tracks") détectées par un système radar. Supposons un système expert qui reçoit les pistes d'avions détectées par les gens d'une section de surveillance et qui a comme mission de les identifier à l'intérieur d'un laps de temps précis. Le système expert donne à l'opérateur l'identification de chaque piste et dans le cas contraire, l'avertit qu'une piste ne peut être identifiée à l'intérieur du laps de temps permis. Notre système expert est un système ouvert sur le monde extérieur puisqu'il reçoit les pistes d'un autre système.

Les informations concernant les pistes sont composées du nom de la piste, de l'heure de sa détection, de sa position au moment présent ainsi que de sa vitesse et de son altitude. Le système expert reçoit les nouvelles détections et les nouvelles positions des pistes déjà détectées à intervalle régulier. Le système expert doit donc être en mesure de faire le suivi d'une piste qui se déplace. Pour ce faire, il a besoin de conserver un historique pour chaque piste. Il doit également faire une distinction entre le traitement d'une nouvelle piste et le traitement d'une piste déjà dans le système. Le système expert doit donc traiter simultanément plusieurs informations de type piste, et ce à l'intérieur du délai fixé à l'avance qui s'applique à chacune des pistes. Certaines des informations reçues portent sur de nouvelles pistes alors que d'autres sont des mises-à-jour des pistes existantes.

Une stratégie adaptée
à une restriction de type temporel

Dans l'application qui nous intéresse, il serait inapproprié d'utiliser un système expert conventionnel dans lequel les pistes d'avion seraient identifiées successivement. Dans ce genre de système, puisque les informations relatives à une piste sont reçues à intervalle régulier, toutes les autres pistes détectées devraient attendre que la piste traitée soit complètement identifiée avant de commencer l'identification d'une autre piste. Ce genre de système est inacceptable du fait qu'il est possible que des pistes n'aient pas la chance d'être identifiées avant l'expiration de leur délai d'identification. De plus, il serait sans intérêt de limiter le système expert à identifier une seule piste alors qu'il aurait du temps disponible pour en traiter plusieurs.

Étant donné que toutes les informations pour identifier les pistes sont obtenues sur un intervalle de temps, le système expert doit pouvoir garder un historique de chacune des pistes qui ont été détectées et qui n'ont pas encore été identifiées. Il va de soit que le système expert doit pouvoir mener de front l'identification de plusieurs pistes. Le problème qui apparaît à ce stade est de déterminer quand et comment le système expert doit traiter chacune des pistes. Regardons donc le problème plus en profondeur.

Chacune des pistes reçues par le système expert doit être identifiée à l'intérieur d'un temps fixe. On peut donc penser que les pistes les plus vieilles devraient être traitées en premier puisqu'il reste moins de temps pour identifier ces pistes que celles plus récentes. La stratégie de l'âge des faits contenus dans les règles de l'ensemble de conflits semble, à premier abord, applicable. Lors de l'utilisation de cette stratégie, le moteur d'inférence choisira toujours les règles dont les faits sont les plus vieux donc les plus vieilles pistes. Jusqu'ici tout semble parfait. Les problèmes arrivent lorsque pour identifier une vieille piste, on infère de nouveaux faits lesquels permettront éventuellement d'identifier la piste. Puisque ces nouveaux faits sont récents, ou si vous préférez moins vieux que d'autres pistes, le moteur d'inférence arrêtera d'essayer d'identifier notre première piste, laquelle est maintenant représentée par de nouveaux faits. Le système expert s'attaquera à d'autres pistes qui sont plus jeunes que la vieille piste initiale mais plus vieilles que les faits inférés. La stratégie reliée à l'âge des faits contenus dans les règles de l'ensemble de conflits n'est donc pas tout-à-fait approprié. La stratégie de l'âge des règles de l'ensemble de conflits est elle aussi soumise aux même problème.

Le principe d'utiliser l'âge des faits contenus dans les règles de l'ensemble conflits est approprié lorsqu'on choisit les pistes qui doivent être identifiées en premier. Il faudrait que tout nouveau fait impliqué dans le processus d'identification d'une telle piste soit considéré avec la même priorité que la dite piste. On peut donc concevoir un degré de priorité qui serait associé à chacun des faits. Ce degré de priorité serait fonction de la piste à identifier et serait partagé par tous les faits impliqués dans le processus d'identification de cette même piste. Le degré de priorité associé aux faits serait la première stratégie à appliquer lors de la sélection de la règle à exécuter. Ainsi, toutes les règles impliquées dans le processus d'identification d'une même piste d'avion auraient le même degré de priorité des faits. Ces règles, par le fait même, seront traitées en fonction du degré plus ou moins élevé de leur priorité de faits. Toutes les autres règles ayant un degré de priorité moindre seront traitées lorsqu'il n'y aura pas de degré de priorité de faits qui leur seront supérieur dans l'ensemble de conflits.

Une combinaison de stratégies adaptée à une restriction de type temporel

Nous venons donc de définir la première stratégie de résolution de conflits à utiliser avec une restriction de type temporel. Cette stratégie n'est cependant pas suffisante dans le cas où l'ensemble de conflit contient plus d'une règle traitant des faits ayant un même degré de priorité. Par exemple, plusieurs règles différentes peuvent s'appliquer aux même faits dans le but de vérifier des aspects différents de l'historique d'une piste. Dans ce cas, nous avons besoin d'une deuxième stratégie. Étant donné que la première stratégie veille à ce que les données prioritaires soient traitées en premier, nous pouvons concevoir qu'il serait adéquat d'utiliser une stratégie permettant de tenir compte de l'importance des règles lors du déroulement d'un raisonnement. Ainsi la stratégie du degré de priorité associé aux règles semble tout indiqué pour devenir la seconde stratégie.

Comme la combinaison de ces deux stratégies ne donne pas l'assurance que seulement une règle sera choisie dans la phase de sélection, une troisième stratégie doit être ajoutée. La troisième stratégie sera celle du "hasard" afin d'avoir la certitude qu'une seule règle soit sélectionnée.

Conclusion

Selon le type d'application qui nous intéressait, une nouvelle stratégie de résolution de conflits, appelée le "degré de priorité associé aux faits", a été définie. La caractéristique principale de cette stratégie est de veiller à ce que, pour certaines données dont le temps de traitement est limité, le traitement associé aux plus vieilles données est exécuté avant celui associé aux plus jeunes. Par exemple, l'identification des vieilles pistes sera exécutée en priorité puisque pour ces dernières, il reste moins de temps de traitement disponible.

La combinaison de stratégies proposée pour l'application de l'identification des pistes permet de tenir compte de plusieurs aspects de résolution de ce problème. Premièrement, la stratégie du "degré de priorité associé aux faits" permet de tenir compte de l'importance des faits. Ainsi le traitement des faits les plus prioritaires sera toujours effectué en premier lieu. Deuxièmement, la stratégie du "degré de priorité associé aux règles" permet de tenir compte de l'importance des règles. Ainsi, lors de l'identification d'une piste, les aspects les plus critiques seront vérifiés en premier. Et troisièmement, la stratégie du hasard permet de faire la sélection d'une seule règle lorsque plus d'une règle a été sélectionnée par les deux autres stratégies.

En bref, la stratégie de résolution de conflits du "degré de priorité associé aux faits" permet d'abord et avant tout de traiter les faits selon l'importance qu'on doit leur accorder. Cette stratégie peut être utilisée dans toute application où certains types de faits doivent être traités à l'intérieur d'un temps donné. Par contre, la combinaison de stratégies proposée ne résoud pas tous les problèmes occasionnés par une contrainte temporelle. Aussi, un aspect qui pourrait faire l'objet d'investigations futures est la gestion du manque de temps d'exécution.

Références

1. M. Bélanger, "Real-Time Expert Systems", *DREV M-3130/93*, avril 1993, SANS CLASSIFICATION.

2. A. C. Stylianou, G. R. Madey et R. D. Smith, "Selection Criteria for Expert System Shells: A Socio-Technical Framework", *Communications of the ACM*, vol. 35, n° 10, p. 30-48, octobre 1992.

903158

Real-Time Expert System Shells: Evaluation Criteria and Results for Commercially Available Tools

Richard G. Brown[1]
Software Kinetics Ltd.
Stittsville, Ontario, K1S 1E7

Micheline Bélanger[2]
Defence Research Establishment, Valcartier
Courcelette, Quebec, G0A 1R0

#149645

Abstract

Seven categories of criteria were developed for evaluating Real-Time Expert System (RTES) shells. The categories included more than three hundred criteria and encompassed system requirements, development and run-time environments, and knowledge base and inference engine capabilities. Next, five RTES shells were evaluated, the prices of which ranged from approximately one to more than fifty thousand dollars. The two least expensive shells did not provide adequate features for real-time programs but were still potentially useful because key components could be combined with suitable external functions; the other three provided many useful features. Unfortunately, the performance of run-time systems could not be compared in part because they were designed to operate in disparate run-time environments. The above work was performed for Defence Research Est., Valcartier by Software Kinetics Limited.

Introduction

A real-time expert system (RTES) is an expert system (ES) which considers the passage of time important as it acquires and processes data. Obviously, the length of time to process data is important in any system; however, in a typical real-time application, inadequate system performance can result in lost input data and/or results which are produced too late to be useful. This contrasts sharply with non-real-time applications in which poor performance usually results in little more than delayed results and user aggravation. Consequently, RTES shells must provide developers with software development tools not normally required for non-real-time applications. Since typical RTES shells are large, complex pieces of software, it was decided that evaluation criteria were required to help developers choose the most appropriate shell.

Evaluation Criteria

Seven categories of criteria were created, some of which were found in a review of criteria for non-real-time shells [Stylianou, et al.]. Others were based on RTES characteristics as described by [Diaz] and [Laffey, et al.] and the recommendations of a real-time application expert. Due to space limitations, only the Inference Engine Criteria of Table IV are presented in full; others tables list section titles only. Criteria related to real-time issues are distributed throughout, many appearing in Tables III and V. Copies of the complete tables may be obtained from the authors.

Table I contains more than fifty criteria describing hardware and software requirements for development and run-time environments. Common platforms are listed (e.g., PCs and Unix workstations), various operating systems, and the need for additional software tools such as text editors and C compilers.

Table II contains more than fifty criteria related to the development environment. These include the type of RTES produced (i.e., a standalone program or data files interpreted by a run-time system), multiple programs, a graphical user interface (GUI), and a GUI builder for end-users. Included text editor features are listed as well as support for external languages (e.g., C or Prolog), and tools for importing data from graphic

---

[1] Phone (613) 831-0888.

[2] Phone (418) 844-4379.

Table I: System Requirement Criteria

1.1	Development Environment Requirements
1.1.1	Hardware
1.1.1.1	Architecture
1.1.1.2	Memory Requirements
1.1.1.3	Distribution Media
1.1.1.4	Vendor Software Security Features
1.1.1.5	Hardcopy Support
1.1.2	Software Requirements
1.1.2.1	Operating System
1.1.2.2	Additional Software
1.1.3	Portability Issues
1.2	Run-Time Environment Criteria
1.2.1	Hardware Requirements
1.2.1.1	Architecture
1.2.1.2	Memory Requirements
1.2.1.3	Vendor Software Security Features
1.2.1.4	Hardcopy Support
1.2.2	Software Requirements
1.2.2.1	Operating System
1.2.2.2	Other Run-Time Requirements
1.2.3	Portability Issues

files or spreadsheets. Documentation criteria include the page count, examples, and bug and error message lists. Debugging criteria include static means (e.g., a variable name-rule cross index) and dynamic means (e.g., debugging using simulated real-time data).

Table III contains seventy-five criteria describing knowledge base (KB) features. These include logic precision, modularization schemes (e.g., hierarchical or block structured), variable data types, variable time histories, use of "facts" (some RTES store data in variables, objects, etc.), rule form (e.g., IF-THEN or IF-THEN-ELSE), rule right-hand side (RHS) contents (i.e., a predicate list or procedural code, the latter preventing backward chaining), and logical operators (including temporal logic operators). Frames, objects, and procedural code are listed as knowledge representation paradigms, and meta-knowledge criteria include multiple KBs, dynamic rule-KB associations, focus of attention mechanisms, etc.

Table IV is shown in its entirety to demonstrate the typical level of detail in a table; here, almost forty criteria describe inference engine capabilities. These include basic inferencing modes such as forward or backward chaining, search strategies (breadth- and depth-first being the most well known), search termination criteria, conflict resolution (CR) options, truth maintenance system support, and temporal control mechanisms. The latter specify when rules should be

Table II: Development Environment Criteria

2.1	Main Features
2.1.1	Type of Application Produced
2.1.2	Programs
2.1.2.1	Interactive Utilities
2.1.2.2	Command-Line Utilities
2.1.3	Included Text Editor
2.1.3.1	Availability
2.1.3.2	Style
2.1.3.3	Features
2.2	Integration Facilities
2.2.1	Embedded External Language Support
2.2.2	Tools to Import Data
2.3	Documentation
2.4	Debugging Facilities
2.4.1	Application Modules
2.4.2	Simulated Input Data
2.4.3	Static Debugging Tools
2.4.4	Simulated Run-Time Debugging Tools
2.5	Backup and Version Control Utilities
2.6	Source Code Security Features

Table III: Knowledge Base Criteria

3.1	Knowledge Representation Paradigms
3.1.1	Logic Precision
3.1.2	Modularization Schemes
3.1.3	Variables
3.1.3.1	Types
3.1.3.2	Value
3.1.3.3	Time Stamps
3.1.3.4	Static and Dynamic Criteria
3.1.4	Facts and Rules
3.1.4.1	Facts
3.1.4.2	Rules
3.1.4.3	Logical Operators
3.1.4.4	Closed World Assumption
3.1.5	Frames
3.1.5.1	Class Hierarchy and Inheritance
3.1.5.2	Class Attributes
3.1.5.3	Static and Dynamic Criteria
3.1.6	Objects
3.1.6.1	Main Features
3.1.6.2	Dynamic Criteria
3.1.7	Other Paradigms
3.2	Mathematical Functions
3.3	Meta-Knowledge Features

tested, facts should be created or deleted, and functions should be called. A number of confidence factor models are also listed.

Table V describes the run-time environment using fifty criteria. It is assumed that an application can run continuously without supervision and that no significant pauses for "garbage collection" will occur (garbage collection is often encountered in LISP-based systems). Criteria describe means of handling hardware and software interrupts (which usually depends on the target hardware and operating system), the number of processes in an application, and support for interprocess communication (IPC) using pipes, shared memory, etc. End-user interface features are listed as well as explanation facility characteristics. Robustness criteria indicate if a system is capable of restoring itself to a known state and continuing after failure, re-establishing failed communication links, and saving system parameters in case run-time errors are detected. Listed performance measures include rule firing rate limitations, bus and main memory loading, and the ability to respond given time constraints.

Table VI uses seventeen criteria to describe archive and playback criteria. Typically, data is archived so that system behaviour can be studied at a future point in time. Besides the ability to record variable values and operator inputs, criteria indicate if fired rules can be recorded and if data may be recorded periodically, on the occurrence of events, or on operator demand. Possible file formats are also listed. Playback criteria indicate, among other things, if archived data may be used as test data during program development and if it can be converted for use by off-line programs such as spreadsheets.

Table VII contains more than twenty-five vendor-related criteria. These include software costs (e.g., evaluation copy, development and run-time system costs) and training course costs. Product support criteria include upgrade policy, support duration, and means of support (e.g., phone, FAX, and e-mail). Other criteria describe licensing details, how long the shell has been available, if the shell is a major product of the developer, and what applications it has been used for.

Evaluation Ratings

In order to use the criteria, a number of qualitative and quantitative ratings were used. 'S' was used to indicate that features were supported without any special development effort. 'CS' indicated that some form of unusual action was necessary to implement or make use of a feature, such as the writing of external C functions (when C was not normally used during program development). An 'NS' rating indicated that the feature was not supported nor practical to implement, such as the addition of an explanation facility or the development of a new conflict resolution mechanism when the inference engine could not be modified. 'Req'd' and 'Not Req'd' were used to indicate that the criteria items were or were not required; for instance, 'Req'd' would be used to indicate that developers had to supply their own C

Table IV: Inference Engine Criteria

4.1	Reasoning Mode
	Forward Chaining
	Backward Chaining
	Bi-Directional Chaining
	Other
	User-selectable
4.2	Search Strategy
	Breadth First
	Depth First
	Rete Algorithm
	Branch and Bound
	Generate and Test
	Best First
	Other
	User-selectable
4.3	Search Termination Strategy
	Stop after First Solution Found
	Stop after All Solutions Found
	User-Selectable
4.4	Conflict Resolution
	User Assigned Rule Priority
	Order of Appearance
	Complexity or Simplicity
	Recency
	Random Selection
	Other
	User-selectable
4.5	Truth Maintenance Systems
	Removal of Unsupported Knowledge
	Removal of Outdated Knowledge
	Consistency Checking
4.6	Temporal Control Mechanisms
	Whenever
	Until
	Every X Time Units
	For X Time Units
	Other
	Minimum Time Increment
4.7	Confidence Factors
4.7.1	Model
	Certainty Factor Model
	Bayesian Probability Theory
	Dempster-Schafer Theory
	Fuzzy Logic Theory
	Other
4.7.2	User Control
	Selectable Model
	Certainty Threshold

compiler. Numeric values were used whenever possible to quantify ratings (e.g., shell memory requirements), and '?' was used to indicate that a rating could not be given because of insufficient data. 'See Below' was also used to direct readers to following comments when none of the above applied.

Besides the ratings, short comments were inserted after each subsection to clarify important points and to describe shell features which could not be adequately described with a simple rating. The comments also helped to put shell features into the larger context; they were invaluable during the shell evaluations.

Evaluated Shells

Five commercially available RTES shells were tested using the evaluation criteria; they were CLIPS, OPS/83, Pamela C, RTworks, and G2. Shell vendors are listed in Appendix A. A very brief summary of the shell evaluations and comparison follows. Note that the systems were tested on a Sun SPARCstation and that the comments may not apply to versions for other platforms. In general, the systems are listed in order of increasing price although G2 and Rtworks can be very similar, depending on the configuration.

Table V: Run-Time Environment Criteria

5.1	Main Features
5.1.1	Continuous Running
5.1.2	Processes
5.2	Integration Facilities
5.2.1	Interprocess Communication
5.2.2	Network Protocols Supported
5.3	End-User Interface
5.3.1	Main Features
5.3.2	Look and Feel
5.3.3	Explanation Facilities

5.3.4	Graphic Displays
5.4	Robustness
5.4.1	Automatic Restart
5.4.2	Process Errors
5.4.3	Run-Time Error Detection
5.5	Performance Criteria
5.5.1	Latency
5.5.2	Loading
5.5.3	Best Guess Given Time Constraints
5.6	Application Software Security Features

CLIPS Version 5.1

The first version of CLIPS (C Language Integrated Production System) was developed at NASA in 1985. From the outset, CLIPS has been designed to be embedded and so it is very useful for incorporating expert systems into real-time programs.

System requirement criteria for Version 5.1 indicate that CLIPS provides a platform-independent command-line "development system" for creating and interpreting small non-real-time expert systems. Windowed development environments are also available for many popular computers (PCs, VAX, workstations, etc.), all of which provide essentially the same facilities as the command-line version. Unlike any other shell tested, CLIPS comes with full C source code for the command-line version and so it may be recompiled for use on other computers.

CLIPS development environments cannot produce standalone applications and they do not support features which are necessary in real-time programs. However, since full source code is provided along with many KB and inference engine interface functions, it is very easy to embed a CLIPS expert system into a real-time application. A typical approach would make use of an infinite loop to acquire data, assert it into the KB, and invoke the inference engine, the latter two steps using provided interface functions. The development environments can also transform KBs into C code so that they may be compiled and embedded. Documentation is thorough although two introductory programming guides are written in an annoying style.

CLIPS KBs use LISP-like code for procedural functions, facts, and prioritized IF-THEN rules. The RHSs of rules contain procedural code and so backward chaining is not possible. CLIPS comes with COOL (CLIPS Object-Oriented Language), a true object-oriented environment; however, objects cannot be used on the left-hand sides of rules. Unfortunately, no means of partitioning knowledge bases into modules is available. CLIPS forward chaining inference engine offers a number of conflict resolution functions. Its only concept of time is the recency of asserted facts. CLIPS has no archive or playback facilities.

CLIPS costs approximately US $1000 when purchased from the official distributor. However, it is sold without license restrictions and so third party vendors have been known to resell it for less than $50. Its portability, flexibility, and relatively low cost have resulted in thousands of CLIPS users.

Table VI: Archive and Playback Criteria

6.1	Archive Criteria
6.1.1	Recordable
6.1.2	Conditions
6.1.3	File Format

6.2	Playback Criteria
6.2.1	Simulated Real-Time Measurement
6.2.2	Conversion Facilities

Table VII: Vendor-Related Criteria

7.1	Price Criteria
7.1.1	Software Costs
7.1.2	Training Courses
7.2	Product Support
7.3	Licensing
7.4	Vendor Commitment

OPS/83 Version 4.0

OPS/83 is a descendent of the OPS series of languages which were developed at Carnegie-Mellon University by Dr. C.L. Forgy (the developer of the RETE algorithm) and others. System requirement criteria indicate that OPS/83 is available for PCs, VAX, workstations, etc. Users must provide their own text editor and C compiler.

The development environment produces stand-alone, single-process applications. OPS/83 source code is written with a text editor and a 'makefile' is used to convert it into C and to compile it. An X Window debugger may be used to test the results. OPS/83 does not provide any real-time program features; however, its inference engine and knowledge bases can be embedded. Typically, a main processing loop (written in C) would be used to continually acquire data, assert it into the KB, and invoke the inference engine. Overall, documentation is adequate but some sections, e.g., how to access an embedded KB, are weak.

OPB/83 KBs consist of one or more modules, each containing data structure definitions, variable declarations, prioritized IF-THEN rules, functions and procedures. Procedural knowledge is implemented in imperative code unique to OPS/83 or in external C functions. KBs may contain multiple, dynamically created "working memories" (WMs), each of which may be associated with rules in various modules. The forward chaining inference engine offers three CR functions and it supports the use of "generalized forward chaining" (GFC). It is claimed that GFC can approximate backward chaining and it can simplify KB development significantly. Users may also define their own recognize-act and conflict resolution functions.

Since OPS/83 lacks features for real-time applications, many run-time criteria are not relevant. User interface facilifies are limited to printed text and command-line prompts although third party GUI libraries could be used. Archive and playback facilities are not provided.

OPS/83 prices range from US $1950 to $50,000, depending on the computer. The Sun version tested costs US $6000.

Pamela C Release 1.4.1

Pamela C (**P**attern **M**atching **E**xpert System **La**nguage) has been developed by Alcatel Austria-ELIN Research Centre. System requirement criteria indicate that it is available for Sun, PC, PC Transputer, and VAX computers. Users must supply their own text editors and C compilers.

The Pamela C development environment consists of command-line or Motif-based programs, the latter providing an integrated development environment (IDE). The IDE provides numerous useful features such as a graphical display of rule dependencies, but some are awkward to use and bugs were encountered much too frequently. Pamela C code is a superset of C and so external C libraries may be used easily. The documentation is well organized and concise.

Knowledge bases are written in Pamela C code, a language which may be described as a combination of (1) C (for implementing procedural knowledge), (2) the non-procedural components of OPS/83 (i.e., WMs, rules with procedural code on the RHS, etc.), and (3) time-based extensions to the above. Programs may contain multiple KBs, rule groups may be selectively activated, and demons (a kind of rule which fires virtually immediately) can be used to handle high priority events. Real-time features include the ability to request the creation or deletion of WM "elements" (facts) at future points in time, to schedule function calls, to suppress actions such as demon firing by specifying critical regions, and to handle asynchronous events using a prioritized list of queues for each KB. Users may also define separate rule and demon CR functions for the forward chaining inference engine.

Like CLIPS and OPS/83, Pamela C does not provide convenient run-time facilities such as network support, GUIs for end-users, IPC functions, or archive and playback facilities. User interfaces are limited to printed text and command-line prompts unless GUI code is developed or third party libraries purchased.

Pamela C prices range from roughly Can. $15,500 to $16,300 for VAX and Sun windowed development versions. The Sun system tested costs approximately US $12,000.

RTworks Version 2.1

RTworks is used to create high-end RTES for use on workstations and VAX computers. An RTworks application consists of multiple processes; typically, this includes an RTserver for handling all IPC and one or more of the following: RTdaq (data acquisition) processes, RThci (human-computer interface) processes, RTie (inference engine) processes, and RTarchive and RTplayback processes. Run-time system requirements are application dependent; development systems require 16 Mb of memory, 35-122 Mb of disk space (platform dependent), a suitable window environment, and a user-supplied text editor.

The RTworks development environment creates files for the run-time processes. Except for the lack of support in creating many text files, the development environment is excellent. The best facility is RTdraw, a Motif-based program for creating GUIs for end-users; an import facility for DXF-format files is also available. All RTworks processes may be customized with C functions and many processes can be embedded into external programs. Networking facilities are also provided which permit run-time processes to be distributed over a non-homogeneous network. Over 2300 pages of excellent documentation is provided.

RTworks KBs offer many features for real-time applications including frames, frame slot time histories, time-triggered IF-THEN-ELSE rules, dynamic rule priorities and priority threshold levels, overlapping rule groups which may be selectively activated, time-based statistical and integration functions, callback functions, etc. The inference engine has both forward and (limited) backward chaining capabilities.

Many useful run-time facilities are provided. For instance, processes have the ability to restart others which have failed and to re-establish failed communication links. Archive and playback facilities are also excellent. For instance, archived data can be used to preload variable time-histories upon process initialization and archived data can be accessed on-line.

Bundled RTworks development and run-time systems cost US $41,000 and $12,000, respectively. Single run-time processes start at US $1500.

G2 Version 3.0

G2 provides a single process, interpretive environment for creating and running RTES on workstation and VAX computers. Development systems require 24 Mb of memory and 50 Mb of swap space; run-time systems may require less.

G2's development environment is very productive because it is an interpreter. For instance, it permits changes to be made to variable values and rules at run-time. It also makes extensive use of its GUI by providing every type of G2 object (procedure, rule, class definition, instance, etc.) with a graphical representation. G2 has a powerful context-sensitive editor which permits keywords to be chosen from a continually updated list and its English-like rules are the most readable of any system tested. GUIs can be created for end-users and Japanese and Korean character sets may be used. A simulator is provided for generating variable values both on- and off-line, and security features may be used to protect knowledge base modules. Unfortunately, the G2 Reference Manual is poorly organized and it lacks a tutorial.

Knowledge base are created in a hierarchy of modules which may be tested independently. KBs are defined in terms of "objects" which represent physical or abstract entities. Object attributes are class dependent; for instance, variable attributes include simulation formulas, data validity intervals, "focal" objects (which may be used to invoke selected rules), and many others. IF-THEN rule attributes include priorities, scan intervals, etc. An unusual knowledge base restriction is a minimum one second value for specifying time intervals (such as a rule's scan rate). The G2 inference engine supports both forward and backward chaining, limited parallel processing, fuzzy logic, and reasoning under time constraints.

A run-time system may consist of a single G2 processes or, depending on what options are purchased, multiple G2 processes distributed over a network. Archive and playback facilities are also available as well as neural network tools, dynamic scheduling packages, means of accessing G2 from PCs, etc.

A full-featured G2 development environment is available from US $24,000 to $42,000, depending on the platform and options. Run-time systems range from US $12,000 to $21,000.

Shell Comparison

Every shell may be considered to be the "best": CLIPS provides excellent value considering its cost, OPS/83's GFC appears to be very useful and it has a very fast inference engine, Pamela C incorporates C code better than any other, RTworks has the best GUI builder, and G2 has the most convenient development environment. For real-time applications, however, Pamela C, RTworks, and G2 have significant advantages. They offer numerous time-based features which are convenient for acquiring data, representing knowledge, processing and archiving time-dependent data, etc. Although CLIPS and OPS/83 can be embedded in real-time programs, implementing time-based features similar to those found in the other three would require a significant development effort. It is also expected that some features would be impossible to duplicate. On the other hand, CLIPS and OPS/83 are much cheaper and so they may still be the most cost-effective solutions for some applications.

It is also interesting to note that none of the shells were designed to operate with a true real-time operating system such as RTOS or Harmony, and no shell provided an explanation facility even though they are common in inexpensive non-real-time shells. This is not to say that all RTES lack explanation facilities because some PC-based shells do have them.

Conclusions

The criteria were found to be very useful for evaluating the shells; they included many related to real-time issues and they went into sufficient detail to highlight important features. However, additional comments were required in all criteria categories to clarify various points. It is believed that more criteria and/or changing the rating structure would not eliminate the need for such comments because (1) no standard metrics exist, (2) it is difficult to categorize large, complex pieces of software, and (3) comments would still be required to describe how shell features relate to each other.

Available RTES shells (for Sun workstations) come in two varieties: those that do not directly support real-time applications but permit key components to be embedded in real-time programs, and true real-time shells which provide numerous features for developing real-time applications. The latter have definite advantages but since they are up to forty times more expensive, the best choice is not obvious.

The authors would like to thank Mr. Selim Ulug, Ms. Cheryl Crabb, and Mr. Ron Lorenson for their contributions to this project.

References

Diaz, A. C., "An Overview of Realtime Expert Systems", NRC Technical Report No. 31759, Division of Electrical Engineering, National Research Council Canada, April 1990.

Stylianou, A. C., Madley, G. R. and Smith, R.D., "Selection Criteria for Expert System Shells: A Socio-Technical Framework", *Communications of the ACM*, Vol. 35, No. 10, pp. 30-48, Oct. 1992.

Laffey, T. J., Cox, P. A., Schmidt, J. L., Kao, S. M. and Read, J. Y., "Real-Time Knowledge-Based Systems", *AI Magazine*, Spring 1988, pp. 27-45.

Appendix A. Shell Vendors

CLIPS: COSMIC
The University of Georgia
382 East Broad Street
Athens, GA 30602-4272
Phone: (706) 542-3265
FAX: (706) 542-4807

OPS/83: Production Systems Technologies
5001 Baum Blvd.
Pittsburgh, PA 15213
Phone: (412) 683-4000
Fax: (412) 683-6347

Pamela C: Alcatel Austria Forschungszentrum GmbH
Ruthnergasse 1-7
A-1210 Vienna, Austria
Phone: (431) 39 16 21
FAX: (431) 39 14 52

RTworks: Talarian Corporation
Suite 400
8410 W. Bryn Mawr Avenue
Chicago, IL 60631
Phone: (312) 380-1223
FAX: (312) 380-1247

G2: Gensym Canada Ltd.
Suite 201
871 Victoria North
Kitchener, Ontario
N2B 3S4
Phone: (519) 741-3680
FAX: (519) 741-3623

391233

SUBSEA ROBOTICS RESEARCH AT OERC/MUN

Michael Hinchey
Ocean Engineering Research Centre
Memorial University of Newfoundland
St John's Newfoundland Canada A1B-3X5

#149647

Abstract

This paper presents an overview of work being conducted by the subsea robotics group in the Ocean Engineering Research Centre at Memorial University of Newfoundland (OERC/MUN) in Newfoundland Canada. Several rigs being developed are described and requirements for control are discussed.

Introduction

About three years ago a subsea robotics group was formed at OERC/MUN. It was formed mainly to allow local students to do graduate research in the subsea robotics area. Student projects are chosen mainly for their educational potential and all have an experimental focus. Our long term goal is to develop a series of highly intelligent subsea robots suitable for various offshore tasks. This paper describes the efforts of our group to date.

Classical control strategies use drive signals based on state error or equations of motion to make a system home in on set points. Supervisory strategies try to mimic a human operator and usually operate at a higher level than classical strategies. For example, they sometimes decide when control by classical schemes is likely to succeed and the set points and gains for it. Supervisory schemes are recent developments. The paper tries to give the essence of each for low level subsea robot tasks. It is hoped that the presentation will give readers who are not control experts a basic understanding of the strategies. Such is hard to get from the literature due to overuse of jargon and/or lack of explanation.

SUBSEA ROBOT CONTROL

An integral part of most modern controllers is the digital computer which takes in signals from various sensors and gives out signals to various drives. When designing such a controller, the main concerns are : stability, accuracy, robustness and computational burden. Transients generated by sudden disturbances gradually decay to tolerable levels when a system is stable: when it is unstable, they grow to dangerous levels. Obviously, to be of any use, a subsea robot must be stable. Accuracy refers to the difference between what a system is doing and what it should be doing. For subsea robots used for things like rig inspection, accuracy would be important. Robustness refers to the ability of a controller to make a system perform a task even when there is uncertainty in the system or its surroundings. Robustness would be important for subsea robots trying to operate during storms or near subsea structures. Computational burden refers to the time required to perform calculations within a computer control loop. Because of transport lags and signal sampling and reconstruction phenomena, a long loop time can degrade stability and other performance measures. Computer speeds are now such that the above is not a problem.

Classical control strategies include: pure error-driven proportional / integral / derivative (PID) schemes, error-driven switching procedures, and schemes based on the governing equations such as computed load and suction or sliding mode (Ogata 1990). Pure error-driven or closed loop control generates drive signals based on state error. Computed load control is basically a form of open loop control and is based on the system equations of motion. These give the drive loads corresponding to desired motions and control signals are sent to the drives in an attempt to generate the appropriate loads. Complex computations are often required for this control strategy making it generally computationally expensive. Alone, it is not expected to be accurate for subsea robots because in the equations of motion parameters are not known exactly. Sliding mode control combines a form of computed load with a switching type of error-driven compensation (Slotine 1984). The switching counteracts uncertainties and is said to make control more robust. It ensures that there is always some measure of control because, when uncertainties have known upper bounds, it is able to nudge the system back towards but never exactly to the command state. Yoerger, Cooke & Slotine (1990) have used it for submersible control. To get robust control, estimates of the uncertainty must be reasonable: such an estimate may be hard to get for hydrodynamic loads. If good estimates of uncertainty are unknown, sliding mode seems like a lot of work for nothing. It might be better to simply trial & error tune a combination of computed load and error-driven compensation. Sliding mode for complex uncertain systems may be a waste of time.

Supervisory schemes can often be found at the top of a layered structure connecting sensors and drives. They have been used to make small submersibles operate autonomously in the unstructured subsea environment (Yoerger 1990). When top layers feed into bottom layers, layers form a hierarchy. When layers are independent, they form a heterarchy. Generally, in practice, top layers have a hierarchial structure while bottom layers are heterarchial and often classical. Some bottom layers are finite state machines : meaning they quickly forget past states. A good example of a layered strategy is the subsumption strategy developed by Brooks (1991) and his colleagues at MIT. Subsumption has layers which for the most part operate independently of each other. However, top layers have higher priority and take over or subsume control from lower layers when conditions warrant. For example, a lower layer may try to keep an exploration sub a certain distance above the bottom. However, when the sub approaches something like an underwater cliff a higher layer would take over to get it over the cliff and then pass control back to the appropriate lower layer. So, the surroundings determine the nature of control. Subsumption is said to give robots insect-like intelligence!

To implement supervisory schemes, one often uses one or more of the following : expert shells / fuzzy logic / neural networks.

An expert shell would contain the supervisory experience of one or more human pilots in a set of IF/THEN rules (Harmon & King 1985 and Waterman 1986). Substructures include: knowledge base / context / inference engine. The knowledge base contains the IF/THEN rules. The rules sometimes have a tree-like structure with outputs from one set feeding into a following set. The context starts with known information about the problem at hand. The inference engine connects the base with the context and generates a solution to the problem : the generated information is stored in an expanded context. The simplest inference engine works the known information through each tree path one at a time to see if there are any matches. A priority scheme is used to pick a path from multiple matches.

Fuzzy logic makes use of the concept of a fuzzy set (Cox 1992 and Zadeh 1965). It is said to simulate human thinking by allowing linguistic concepts to be treated mathematically. Like expert shells, it would contain the experience of pilots in a set of IF/THEN linguistic rules augmented by membership functions. Ura (1990) used an elementary version of it to control one of a series of small submersibles being developed in Japan. As noted, like expert shells, fuzzy logic obtains rules from human experts. However, unlike expert shells, it does not require the experts to directly interpolate information. So, experts find it less demanding. Typically, interpolation is done using a simple graphical procedure. In many papers, it is hard to see this due to overuse of set theory and its jargon (Jain 1980). Because fuzzy logic is basically a method of interpolation, it could give very poor predictions outside its data range. Expert shells are known to have the same problem. Note that fuzzy logic has also been used for error driven control. In fact, the earliest application of fuzzy logic was of this type (Mamdani & Sembi 1980).

A neural network would contain the experience of pilots in its weights (Moore 1992, Hammerstrom 1993 and IEEE Press 1992a & 1992b). Yuh (1990) has used it for small submersible control.

When combined with error-driven schemes, fuzzy logic would probably give the most robust control for subsea robots. The fuzzy logic part would try to mimic a human operator and oversee the error-driven part. For example, in severe storms, fuzzy logic could be used to select a safe operating depth and error-driven schemes could be used to home in on it. Inputs into the fuzzy logic would include such things as wave steepness and period: its output would be the safe depth. Each fuzzy logic statement or rule would have a format like "If wave steepness R is large (L) and wave period T is large (L), then safe operating depth D should be very large (VL)". Membership functions would be used to quantify the degree to which a parameter belongs to a particular class like large. The experience of one or more human pilots would be used to generate rules and membership functions. For an expert shell, a pilot would have to estimate the depth D corresponding to specific values of R and T. This would involve some interpolation, either on paper or in his head. The use of membership functions allows him to avoid this. Membership function shapes include triangles, trapezoids and smooth bell shapes. Fig. 1a shows what typical membership functions might look like for the depth/wave case: the thick lines denote the L/L/VL rule. There could be many more classes than the few given here. Generally, neighbouring membership functions overlap. However, here to simplify the presentation, the depth functions are not allowed to overlap. Note that for the inputs denoted by the arrows, R is approximately 0.25 large and 0.75 small while T is 1.0 large and 0.0 small. At sea, R and T would have to be measured somehow. Typically, R would be in the range 0.0 to 0.15 while T would be in the range 0 to 15 seconds. For a specific set of R and T, manipulation of each fuzzy logic rule produces a contribution to an overall fuzzy range of depth. This manipulation for the

L/L/VL rule is given in Fig. 1b and the overall range is shaded black: the contribution of each rule is identified by its number. Note that for the manipulation it is assumed that when a rule has an IF/AND/THEN structure such as each has here the minimum membership value to the left of THEN dominates: when it has an IF/OR/THEN structure then the maximum membership value dominates. Some reflection on this shows that it is just common sense. A crisp depth would be obtained from the overall range using a process known as defuzzification. "Centroid of Area" is a commonly used defuzzification scheme and this is also illustrated in Fig. 1b.

Biological neural networks are made up of many simple processing elements known as neurons connected together in a complex way. In a network, dendrites feed information into each neuron. At each dendrite/neuron interface, a synapse controls the strength of the signal fed in. Synapses are basically memory. Each neuron sums its inputs and fires or sends out a signal along its single axon if the sum is above a certain threshold. The axon branches into dendrites which feed into other neurons. Artificial neural networks try to mimic the biological network. They can be used to form input/output maps or fits.

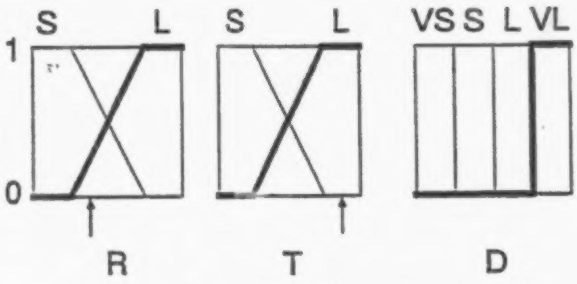
A neural network fit to a depth selector could be used to simplify its implementation. Hopefully, a network trained by a finite number of selector input/output pairs would be able to generalize the logic and thus be able to predict depths for inputs not in the training set. A typical mapping neural network is shown in Fig. 2. It consists of an input layer, a hidden layer and an output layer of neurons. The input and hidden layers each contain one bias neuron with input unity. The summed inputs into hidden layer neurons are processed by a nonlinear squashing function when they pass through the neuron : the sigmoidal or S shape used by most researchers for this is:

$$f(i) = 1 / (1 + e^{-i}) . \qquad (1)$$

Information flows forward through the net from input to output. For a single input / single output case, the mapping equation for a single neuron is:

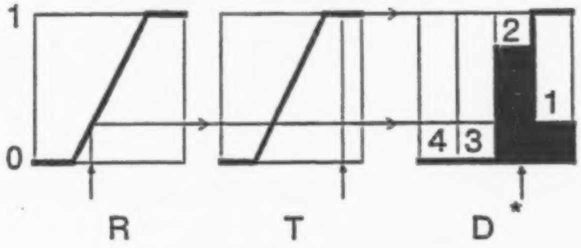
$$O = W_{OB} + W_{OM} \, f (W_{IM} I + W_{IB}) \qquad (2)$$

This equation shows how in an I/O plot the weights can be used to scale and shift the squashing function both horizontally along I and vertically along O. In the general case a map is formed by patching together in hyperspace many scaled and shifted squashing functions. This gives the net a local character something like that of finite elements and this is why good fits are possible.



1) IF **R** IS **L** AND **T** IS **L** THEN **D** IS **VL**
2) IF **R** IS **S** AND **T** IS **L** THEN **D** IS **L**
3) IF **R** IS **L** AND **T** IS **S** THEN **D** IS **S**
4) IF **R** IS **S** AND **T** IS **S** THEN **D** IS **VS**

a) Depth/Wave Membership Functions.



b) Generation of Fuzzy Depth Range and Defuzzification.

FIGURE 1 : Fuzzy Logic Illustration.

For the depth selector case, the goal is to find the weights W_O and W_I which make the net depths O_N match the corresponding target depths O_T. Most neural network codes use steepest descent to adjust the weights in an iterative fashion starting from a random distribution. For depth, one forms the error:

$$e = O_N - O_T \qquad \text{where} \quad O_N = \Sigma\, W_O\, M \qquad (3)$$

where M is the squashed output from a hidden layer neuron. The goal is to minimize e^2. For each input / output training pair the squared error is dependent on each W_O and M feeding into it. For small changes in W_O and M, one can write

$$\Delta e^2 = \Sigma\, [\partial e^2/\partial W_O]\, \Delta W_O + \Sigma\, [\partial e^2/\partial M]\, \Delta M \qquad (4)$$

Now $M = f[\Sigma W_I I]$. Manipulation gives $\Delta M = \Sigma\, \partial M/\partial W_I\, \Delta W_I$. So, the Δe^2 equation becomes:

$$\Delta e^2 = \Sigma\, [\partial e^2/\partial W_O]\, \Delta W_O + \Sigma\, [\partial e^2/\partial M]\, \Sigma\, \partial M/\partial W_I\, \Delta W_I \, . \qquad (5)$$

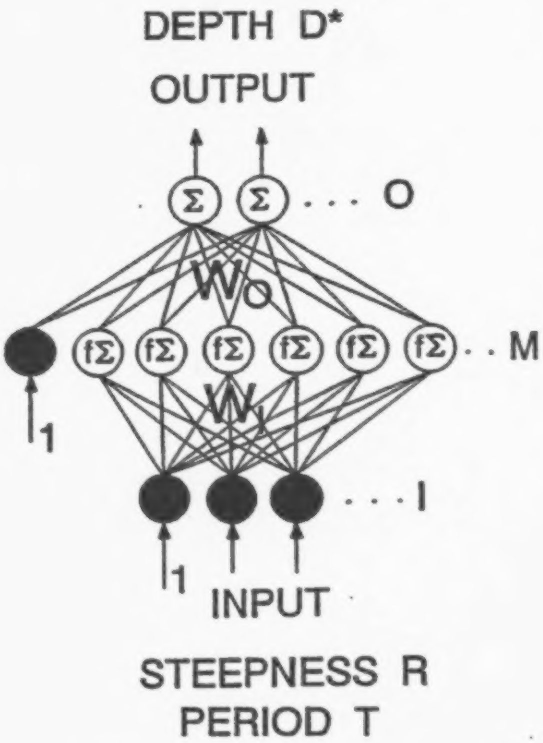


FIGURE 2 : Mapping Neural Network.

According to steepest descent:

$$W_O(NEW) = W_O(OLD) - \kappa\ \partial e^2/\partial W_O$$
$$W_I(NEW) = W_I(OLD) - \kappa\ [\partial e^2/\partial M]\ \partial M/\partial W_I\ . \quad (6)$$

When fed many times with each data point in an input/output training set, this iteration when stable usually backs down the global e^2 versus weight bowl in hyperspace towards the global minimum. A small value of κ promotes iteration stability. To avoid convergence to local minima, noise added to the weights could be used to periodically shake up the iteration.

The mapping neural net could also be used to get a fit to the overall dynamics. This could then be used in a computed load or sliding mode control strategy. The mapping net could also be used for error driven control. In each case, trajectory error could be used to train the net and in each case the response of the system lagged by a step in time would be one of the inputs into the net.

Note that there is another type of neural network which is good for pattern recognition. An example of such a net is that proposed by Kohonen (IEEE Press 1992a & 1992b).

SUBMERSIBLE ESCAPE CAPSULE

During severe storms, evacuation systems for offshore rigs currently in use have proven themselves to be inadequate. Typically, during deployment of a lifeboat, it is often damaged to the point of not being seaworthy before it reaches the ocean surface. This is especially the case for cable-launched boats where a pendulum-like motion of the craft on its cables is often set up. It is less of a problem for free-fall lifeboats. Even when a craft reaches the ocean surface intact, high winds and waves can drive it back against the rig structure. We are developing a small submersible escape capsule for offshore rigs. Presently this consists of a cylindrical centre body with truncated cone end caps. A capsule under construction is shown in Fig. 3. The capsule would be used to evacuate a few people, who, during storms, would remain on board a rig as long as possible and try to ride out the storm. After launch, the capsule would sink below the water surface and thus avoid the violent air and water motions near the surface. Sinking would lower significantly the likelihood of damage due to contact with the rig and also the likelihood of severe or life

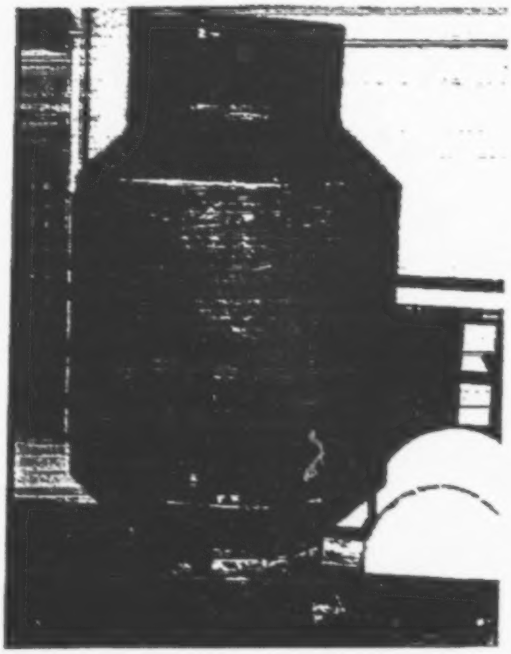


FIGURE 3 : Rig Escape Capsule.

threatening sea sickness. Once below the waves, it would drift with the local current away from the danger zone near the rig. When far from the rig, it would, if water depth is reasonable, anchor itself to the bottom and wait until the storm subsides. In deep water, it would be made to hover at a specific depth below the surface.

For this, a small on board computer with manual override would control air/water ballast automatically. Drop weights would be used to resurface the capsule if the primary control fails.

HYDRODYNAMICS OF SUBSEA ROBOTS

The long term goal is to understand the hydrodynamics of subsea robots with multiple link arms undergoing arbitrary motions far from and close to subsea structures while being subjected to wave/current disturbances. From the work we hope to gain an understanding of the underlying fluid mechanics and how it influences various loads. For early exploratory work, we restricted attention to the drag torque on a single link arm/body configuration undergoing steady body translation and rotation. Recently we examined a two link arm undergoing various combinations of steady translation and rotation. The setup is shown in Fig. 4. During tests, the arm was positioned approximately 1m below the water surface in the MUN wave/towing tank facility: the tank itself being approximately 2m deep. The motion of the towing carriage was used to generate steady translation and DC motors were used to generate rotation. Data from the tests are now being processed.

PNEUMATIC SUBSEA ROBOT

Most subsea robots use ducted propellers for propulsion. Often DC batteries are used to run the propulsion motors. Even when a robot is commanded to hover at a specific depth, these motors have to operate to counteract over or under ballast. In fact, because of the nonlinear characteristics of the system, hovering at a specific depth is very difficult (Yoerger 1990). Typically, craft undergo limit cycle oscillations about the desired depth. For a small subsea robot that we are developing, we are trying to use, instead of motors, a divers gas bottle for depth control: solenoid valves would be used by an on board computer (Z WORLD : LITTLE GIANT) to control flow into or out of a gas/water ballast tank. When developed, we hope that this robot would be something which could be thrown over the side of a ship, go to preset depths or explore the ocean bottom and then return to the surface.

FIGURE 4 : Hydrodynamics Setup.

HYDRAULIC SUBSEA ROBOT

The goal of this project is to see if a gas pressurized servo controlled hydraulic actuator can be used for short duration positioning of subsea robots. Positioning could be accomplished by an actuator through a change in buoyancy or through creation of strong water jets. In the present setup, buoyancy is being controlled with a bellows (Fig. 5). An accumulator is used to store high pressure hydraulic oil. Its size limits the test duration. A cylindrical steel chamber is used to store the oil which flows from the low pressure or return side of the actuator. A Keithley 575 PC Interface together with a EMPaC 386 computer is used to take in signals from the various sensors and give out a control signal to the servovalve. The control signal itself is generated in a C software loop. In a prototype, the controller would be located onboard. However, here, to simplify, it is located externally.

COLLABORATIONS WITH IMD & CSA

The National Research Council Institute for Marine Dynamics (IMD) in Newfoundland recently formed an underwater vehicle group. It is headed by Dr. Indra Datta and its goal is to develop expertise and make it available locally and abroad. Our group is working closely with this group. The Canadian Space Agency (CSA) is a partner in the Space Station Program. Our group believes that Canada on its own should develop a small one person space pod with robot arms to augment systems already chosen for the Space Station. Such a system extends naturally from space robotics work already done in Canada and would be something needed for developments beyond the

Space Station. For routine tasks such as inspection, the pod could operate autonomously: however, for more demanding tasks, such as those beyond or at the limit of space suit and other technologies, it would be manned. To guide development of such a pod, we plan to, with the help of CSA (Dr. George Vukovich), design and build a small one person submarine. We believe that its motions in the Seakeeping Basin at IMD would mimic motions of the pod in space. We recently completed a preliminary design of the sub (Hinchey 1994). Our goal was to establish for the sub its: geometry / propulsion system / neutral buoyancy system / docking system / robot arms. Fig. 6 shows what the sub presently looks like.

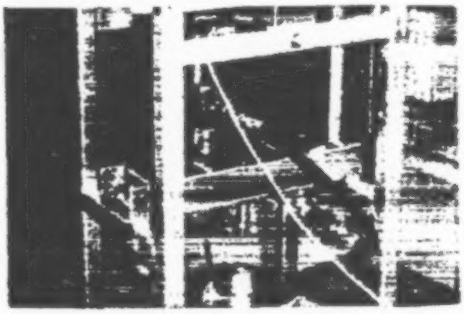


FIGURE 5 : Hydraulic Rig

ACKNOWLEDGEMENTS

This work was funded by the Engineering Faculty Enrichment Fund at Memorial University in Newfoundland and by the Natural Sciences and Engineering Research Council of Canada through Research Grant A4955 (Hinchey). The contributions of graduate students Karen Muggeridge, Carlos Rivera, Yifei Zhang and Zhongqun Lu are hereby acknowledged.

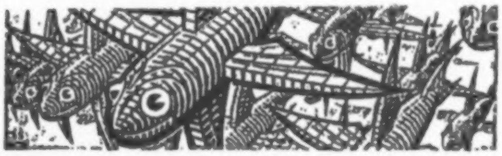


FIGURE 6 : CSA/MUN SUB

REFERENCES

Brooks R. (1991) : "New Approaches to Robotics", Science 253.

Cox E. (1992) : "Fuzzy Fundamentals", IEEE Spectrum (October Issue).

Hammerstrom D. (1993) : "Working with Neural Networks", IEEE Spectrum (July Issue).

Harmon P. & King D. (1985): "Expert Systems", Wiley, New York.

Hinchey M.J. (1994) : "Potential for Subsea Robot Control", Accepted for Publication in the Ocean Engineering Journal.

Hinchey M.J. (1994) : "Development of a Small Manned SPACE POD", CSA Draft Report.

IEEE Press (1992a) : "Neural Networks : Theoretical Foundations and Analysis".

IEEE Press (1992b) : "Artificial Neural Networks : Paradigms, Applications and Software".

Jain R. (1980): "Fuzzyism and Real World Problems", Fuzzy Sets, Plenum Press.

Mamdani E. & Sembi E. (1980): "Process Control using Fuzzy Logic" Fuzzy Sets, Plenum Press.

Moore K. (1992): "Artificial Neural Networks", IEEE Potentials.

Muggeridge K. and Hinchey M. (1992) : "A New Jet Propulsion Device for Subsea Robots", AUV/92.

Ogata K. (1990) : "Modern Control Engineering", Prentice-Hall.

Rivera C. and Hinchey M. (1992) : "Hydrodynamics of Simple Robot Arm/Body Configurations", AUV/92.

Slotine J. (1984): "Sliding Controller Design for Nonlinear Systems", Journal of Control, 40(2).

Ura T. (1990) : "Development of AUV PTEROA", First Workshop on Mobile Subsea Robots, Monterey, USA.

Waterman D. (1986) : "A Guide to Expert Systems", Addison Wesley.

Yoerger D. (1990) : "Precise Control of Underwater Robots: Why and How", First Workshop on Mobile Robots for Subsea Environments, Monterey, USA.

Yoerger D., Cooke J. and Slotine J. (1990) : "The Influence of Thruster Dynamics on Underwater Vehicle Behaviour and Their Incorporation Into Control System Design", IEEE Journal of Oceanic Engineering, 15(3).

Yuh J. (1990) : "A Neural Net Controller for Underwater Robotic Vehicles", IEEE Journal of Oceanic Engineering, 15(3).

Zadeh L. (1965) : "Fuzzy Sets", Information andControl (8).

90C219

Conception d'un mini-véhicule pour l'étude de systèmes de guidage autonome

S. Gill, R. Hurteau, P. Cohen
Département de génie électrique
et de génie informatique
École Polytechnique de Montréal
C.P. 6079, Succ. Centre-Ville
Montréal (Québec)
H3C 3A7

J.-C. Piedboeuf
Mechanical Engineering Department
Royal Military College
Kingston (Ontario)
K7K 5L0

#149C51

Résumé

L'étude et la mise au point de véhicules autonomes, c'est-à-dire capable de percevoir leur environnement, de planifier et d'exécuter leurs tâches, nécessite la réalisation de nombreux essais expérimentaux. Les robots mobiles disponibles sur le marché pour réaliser ces tests sont pour la plupart des plate-formes mobiles, parfois assez encombrantes, et se déplacant à des vitesses lentes. Dans ce contexte, un modèle réduit a été développé à l'École Polytechnique. Ce véhicule est de type automobile, il mesure 47cm par 80cm et peut atteindre une vitesse de pointe de 3 m/s. Différents types de systèmes de vision, caméra monoculaire, caméras stéréo ou télémètre de type BIRIS peuvent être fixés à la tourelle du véhicule. Un modèle dynamique simplifié du véhicule est également présenté et les fonctions de base du contrôleur de mouvement sont discutées.

Introduction

Au cours des dernières années, de nombreux travaux de recherche ont porté sur l'étude et la mise au point de véhicule autonome. Ces véhicules doivent avoir la capacité de percevoir leur environnement immédiat, de planifier leurs tâches et le chemin à suivre et d'exécuter ces mouvements en présence d'obstacles. Ces études impliquent de nombreux essais expérimentaux pour valider les différentes approches proposées et pour évaluer leurs performances. Les robots mobiles disponibles sur le marché, ou développés par les différents groupes de recherche pour réaliser ces tests, sont pour la plupart des plate-formes mobiles [1] ou des véhicules à trois roues [2]. Ces robots mobiles sont en général assez encombrants, se déplacent à des vitesses inférieures à $1m/$sec et ne sont pas représentatifs au niveau cinématique et dynamique des véhicules automobiles ou industriels (véhicule avec remorque, véhicule articulé).

Dans ce contexte, un véhicule automobile de modèle réduit a été développé au département de génie électrique et de génie informatique de l'École Polytechnique. Ce véhicule est de type automobile, c'est-à-dire qu'il est de forme rectangulaire avec des roues motrices à l'arrière et une direction à l'avant: il a donc le même comportement au niveau cinématique qu'une automobile. Ce véhicule est construit selon une échelle de 1:5; il mesure 47cm par 80cm et pèse 32,8 kg. Sa vitesse de pointe est de 3 m/s. Le véhicule est donc assez petit pour être utilisé dans un laboratoire et suffisamment rapide pour qu'il faille tenir compte de la dynamique au niveau du contrôle. Une caméra monoculaire, deux caméras stéréo ou un télémètre laser de type BIRIS [3], peuvent être fixés (1 à la fois selon l'application) à la tourelle du véhicule.

Dans cet article, une description détaillée du véhicule est d'abord donnée. Les caractéristiques et les performances globales sont discutées en les comparant aux paramètres de conception. Un modèle dynamique simplifié du véhicule est ensuite présenté et les fonctions de base de contrôle du mouvement sont discutées.

Description du véhicule

Le véhicule automobile baptisé VIGAR (Vision Guided Autonomous Robot) est montré à la Figure 1. À la conception, les spécifications étaient les suivantes: poids approximatif 35kg, vitesse maximale de 3 m/s pour une accélération maximale de $3m/sec^2$. Le système d'entraînement retenu comprend deux moteurs CC pouvant fournir un couple de $2,5N.m$ à $573rpm$, pour des roues possédant un rayon de $0,05m$. Un coefficient de frottement avec le sol de $0,63N/m.s$ est dans ce cas nécessaire pour qu'il n'y ait pas de patinage au niveau des roues. Le Tableau 1 donne la liste des composants électro-mécaniques du véhicule. Le véhicule expérimental peut atteindre une vitesse maximale de l'ordre de $3m/s$. Cependant son accélération est limitée à 1 m/s^2;ceci est causé par l'alimentation de 24V utilisée au niveau des amplificateurs de puissance.

L'unité de commande informatique embarquée est composée d'un micro-ordinateur 486 de Ziatech (STD32). Le Tableau 2 donne la liste des éléments de ce micro-ordinateur. L'ensemble de ces composants ont été choisis de façon à conserver une architecture

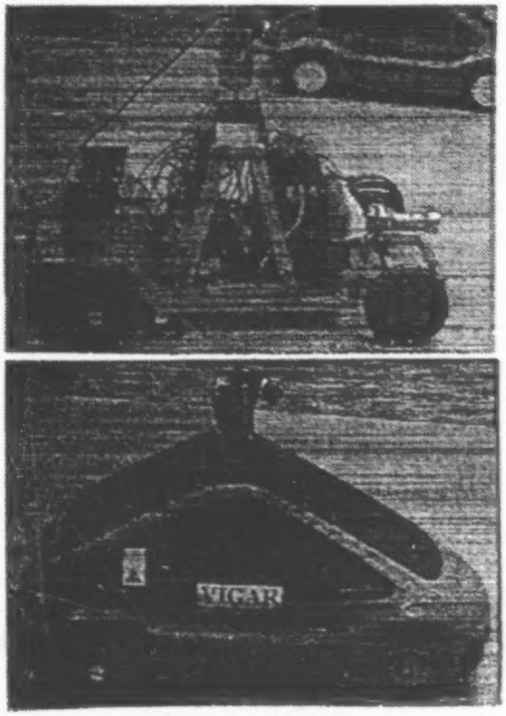


Figure 1: VIGAR

simple et modulaire au niveau du véhicule tout en minimisant le temps de développement et de mise au point.

Le véhicule est relié par deux liens sériels (RS-232C) à un ordinateur-hôte IBM-PS2 et à une station de travail de type SUN SPARC. La Figure 2 montre l'architecture globale du système de commande informatique. Comme il est indiqué sur la figure, le traitement des données provenant de la caméra BIRIS est réalisé sur un ordinateur Datacube et sur une station de travail SUN.

Au niveau de la commande, seules les fonctions de contrôle de mouvement sont implantées au niveau du micro-ordinateur embarqué 486. Le traitement des données télémétriques comme le calcul de la profondeur, la construction de cartes locale et globale, ainsi que les fonctions de planification de chemin sont réalisées sur les stations de travail. La Figure 3 montre l'organisation des différents modules informatiques.

Modèle dynamique et contrôle de mouvement

Le modèle utilisé pour analyser le comportement du véhicule s'inspire du modèle simplifié développé par [4] pour un véhicule à deux roues motrices. Pour ce type de véhicule, on néglige la dynamique reliée au roulis et au tanguage ainsi que les effets associés à la déformation et au glissement des pneus. Ce modèle est alors valable pour des vitesses de déplacements et des accélérations faibles. En tenant compte de ces

Moteurs de propulsions	2 moteurs CC Pittman GM14605, 24Vcc, réducteur 4:1
de direction	1 moteur CC Pittman GM8712G276, 24Vcc, réducteur 96:1
de la tourelle	2 moteurs pas-à-pas SLO-SYN M061-LS08
Amplificateurs	1 Copley Controls amplificateur à modulation de durée modèle 303 2 Copley Controls amplificateur à modulation de durée modèle 306 SLO-SYN Micro Series motion control: 2 modules 430-TH
Capteurs	3 Codeurs optiques HP HEDS-6000, 1000 et 500 imp/tour 3 HCTL-2000 Quadrature decoder / counter interface
Alimentation	2 Batteries GS Portalac PE6.5-12R, 12V, 6.5 Ah 1 Batterie YUASA NP12-12, 12V, 12 Ah convertisseur CC-CC, WR12T05-12/55K-3

Table 1: Composants électro-mécaniques du véhicule

Ziatech ZT-8902	carte CPU 486SX, 20MHz, 4Moctets de mémoires vives
Ziatech ZT-8951	Unité de disquette 3,5 po et contrôleur
Ziatech ZT85CT61	96 entrée-sorties numériques
Sensoray 7405	8 convertisseurs N/A,
Ziatech ZT-200	boîtier STDbus 32 bits,

Table 2: Unité de commande informatique embarquée

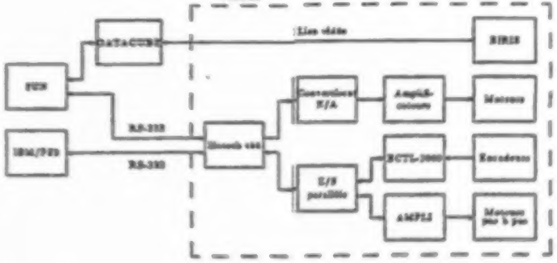


Figure 2: Architecture du système de commande informatique

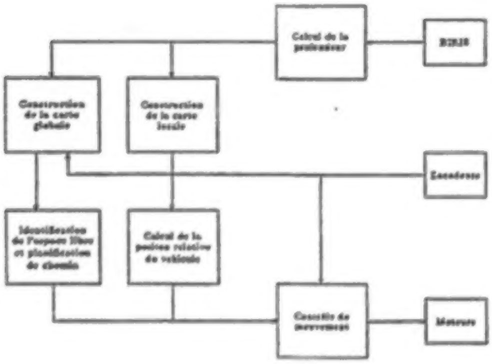

Calcul de la profondeur
Construction de la carte globale
Construction de la carte locale
Encodeurs
Identification de l'espace libre et planification de chemin
Calcul de la position relative du véhicule
Contrôle de mouvement
Moteurs

Figure 3: Modules de commande informatique

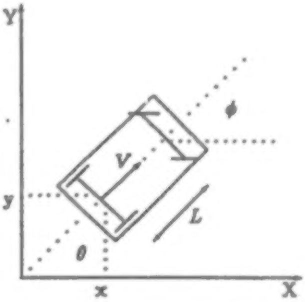

Y
ϕ
V
y
L
θ
x
X

Figure 4: Représentation du véhicule dans le plan XY

hypothèses simplificatrices, le modèle est développé en considérant la cinétique du véhicule auquelle on ajoute le modèle des actionneurs.

À partir du schéma de la Figure 4, on peut écrire les équations décrivant la cinétique du véhicule [5]:

$$\begin{aligned} \dot{x} &= V\cos\theta \\ \dot{y} &= V\sin\theta \\ \dot{\theta} &= V\frac{\tan\phi}{L} \end{aligned} \tag{1}$$

où x: position en x du véhicule,
y: position en y du véhicule,
θ: orientation du véhicule par rapport à l'axe des x,
ϕ: orientation des roues directrices,
V: vitesse de translation du véhicule.

La dynamique du véhicule en translation en fonction des tensions appliquées aux bornes des servomoteurs est obtenue en combinant l'équation de Newton

$$F_1 + F_2 = M\dot{V} + fV \tag{2}$$

et celle du groupe amplificateur-moteur:

$$\Gamma_i = K_{\phi i} K_{ai} U_{tri} \tag{3}$$

De plus

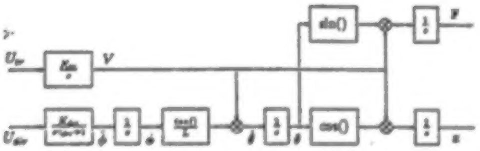

U_{tr}
$\frac{K_{tr}}{s}$
V
$\sin()$
$\frac{1}{s}$
y
U_{dir}
$\frac{K_{dir}}{\tau_{dir}s+1}$
$\dot{\phi}$
$\frac{1}{s}$
ϕ
$\frac{\tan()}{L}$
$\dot{\theta}$
$\frac{1}{s}$
θ
$\cos()$
$\frac{1}{s}$
x

Figure 5: Modèle dynamique

$$F_i = \frac{\Gamma_i}{n_i R_{ai}} \tag{4}$$

avec

U_{tr1}, U_{tr2}: commandes appliquées aux moteurs,
K_{a1}, K_{a2}: gains des amplificateurs ($2amp/volt$),
$K_{\phi 1}, K_{\phi 2}$: constantes de couple des moteurs ($0,0612N - m/amp$),
Γ_1, Γ_2: couples appliqués aux roues,
F_1, F_2: forces de propulsion équivalentes,
R_{a1}, R_{a2} : rayons des roues ($0,07625m$),
n_1, n_2: rapports d'engrenage ($0,25$),
M: masse du véhicule ($32,8kg$),
f: coefficient de frottement visqueux.

En regroupant les termes et en négligeant le frottement visqueux, la dynamique du véhicule en translation est représentés par un intégrateur avec un gain K_{tr} soit:

$$\dot{V} = K_{tr} U_{tr}; \quad avec \quad \begin{aligned} K_{tr} &= \frac{2K_a K_\phi}{nR_a M} \\ K_{tr} &= 0,35m/s^2 - volts \end{aligned} \tag{5}$$

Finalement un modèle du premier ordre est utilisé pour représenter la dynamique de l'actionneur au niveau de la direction du système (le moment d'inertie du véhicule est négligé). On a donc

$$\ddot{\phi} = \frac{K_{dir}}{\tau_{dir}} U_{dir} - \frac{\dot{\phi}}{\tau_{dir}} \tag{6}$$

avec U_{dir}: commande du moteur de direction,
K_{dir}: gain au niveau de la direction,
τ_{dir}: constante de temps au niveau de la direction.

Le modèle complet est représenté à la Figure 5. Les gains K_{dir}, K_{tr} et la constante de temps τ_{dir} ont été identifiés à partir d'essais expérimentaux (réponse à l'échelon) et leurs valeurs sont données dans le Tableau 3. Il est à remarquer que le gain en translation mesuré est très proche du gain théorique. Ce modèle a été vérifié expérimentalement et est valable pour des vitesses faibles (moins de 3 m/s).

Pour réaliser la conception d'un contrôleur de mouvement, le système peut être découplé et linéarisé. En effet, suivant la méthode décrite dans [6] et [7], il faut d'abord exprimer le modèle dans le référentiel du chemin et considérer que l'erreur d'orientation ε_θ est petite. En supposant de plus que $V \approx V_{ref}$ et que $\phi \approx \tan\phi$ ($\phi < 34°$), les modèles découplés s'écrivent alors de la façon suivante:

K_{dir}	8218 $deg/s^2 - volts$
τ_{dir}	3 ms
K_{tr}	0.32 $m/s^2 - volts$

Table 3: Paramètres du modèle mesurés expérimentalement

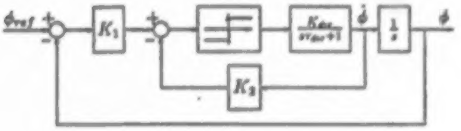


Figure 6: Contrôleur de type tout-ou-rien

le modèle du véhicule en translation:

$$\begin{bmatrix} {}^{e}\dot{X} \\ \dot{V} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} {}^{e}X \\ V \end{bmatrix} + \begin{bmatrix} 0 \\ K_{tr} \end{bmatrix} U_{tr} \quad (7)$$

le modèle du véhicule en rotation:

$$\begin{bmatrix} \dot{\varepsilon_y} \\ \dot{\varepsilon_\theta} \\ \dot{\phi} \\ \ddot{\phi} \end{bmatrix} = \begin{bmatrix} 0 & V_{ref} & 0 & 0 \\ 0 & 0 & \frac{-V_{ref}}{L} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & \frac{-1}{\tau_{dir}} \end{bmatrix} \begin{bmatrix} \varepsilon_y \\ \varepsilon_\theta \\ \phi \\ \dot{\phi} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{K_{dir}}{\tau_{dir}} \end{bmatrix} U_{dir} + \begin{bmatrix} 0 \\ \dot{\theta}_{ref} \\ 0 \\ 0 \end{bmatrix} \quad (8)$$

avec ε_θ et ε_y respectivement l'erreur d'orientation et l'erreur latérale.

Le modèle résultant étant découplé et linéaire, il est possible d'utiliser des contrôleurs linéaires classiques pour l'asservir. Le système en translation est asservi en vitesse par un contrôleur de type PI, selon la loi de commande suivante:

$$U_{tr} = K_{tr1} Err + K_{tr2} \int Err dt \quad (9)$$

avec $Err = V_{ref} - V$;
V_{ref} étant la vitesse désirée

Les valeurs des gains sont choisies expérimentalement de façon à obtenir un temps de réponse de l'ordre de 100 ms. Les valeurs des gains sont K_{tr1}=15 et K_{tr2}=3.

Deux contrôleurs imbriqués sont utilisés pour commander le système en rotation. Un premier contrôleur de type tout-ou-rien assure que l'orientation ϕ des roues directrices est égale à la valeur de consigne ϕ_{ref} (Figure 6). Les gains de ce contrôleur sont choisis expérimentalement de façon à obtenir un temps de réponse de 10 ms.

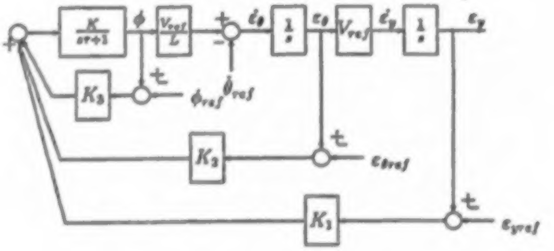


Figure 7: Contrôleur d'orientation par retour d'état

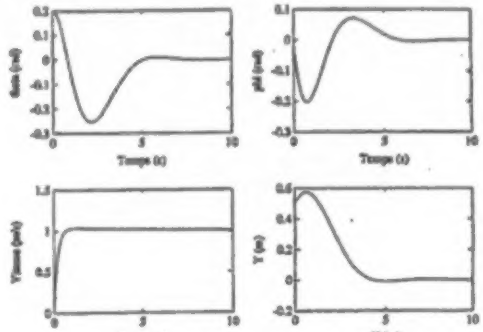


Figure 8: Correction d'erreur latérale et d'orientation

Par la suite, on suppose que la fonction de transfert entre ϕ et ϕ_{Ref} est du premier ordre (avec $K = 1$ et $\tau = 10ms$). La deuxième boucle de contrôle en rotation est alors une boucle de commande par retour d'état tel que représentée à la Figure 7. Les gains K_i de cette boucle de commande sont ajustés en fixant les pôles décrivant la dynamique du système boucle fermée.

Les contrôleurs proposés ont été analysés en simulation. Les résultats obtenus pour la correction d'une erreur latérale et d'une erreur d'orientation sont présentés aux Figures 8. Ces résultats correspondent à des pôles désirés égaux à -0.8317, $-0.7035 + 0.9763i$ et $-0.7035 - 0.9763i$.

Une comparaison entre l'analyse en simulation et les essais sur le véhicule a été réalisée. Les résultats correspondants à la correction d'une erreur latérale sont présentés à la Figure 9. Les trajectoires de l'orientation θ du véhicule et de l'angle de direction ϕ diffèrent légèrement. Cependant, la courbe de l'erreur latérale (Y) en fonction de la distance X indique globalement une bonne correspondance entre le modèle et le système. Des essais complémentaires permettront d'établir le domaine de validité du modèle.

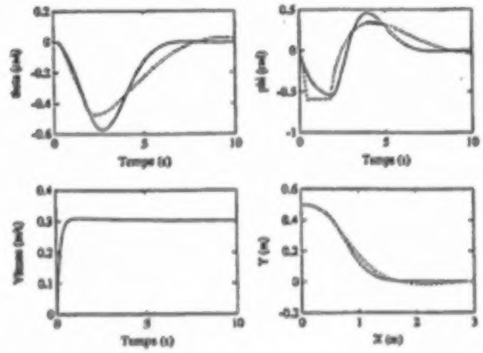


Figure 9: Comparaison entre le modèle et le système

Conclusion

Le véhicule automobile présenté dans cet article est de petite taille et peut se déplacer à une vitesse de pointe de $3m/sec$. Il est muni d'une tourelle pouvant transporter soit une caméra monoculaire, deux caméras stéréo ou un télémètre BIRIS. Le micro-ordinateur embarqué Ziatech 486 permet d'implanter les fonctions de contrôle de mouvement. Les fonctions d'analyse d'image sont réalisées sur les stations de travail de type SUN. Ce véhicule constitue donc un banc d'essai idéal pour étudier en laboratoire les systèmes de guidage autonome. Son faible encombrement, ses performances en accélération et en vitesse et son architecture informatique ouverte en font un outil de travail idéal. Les coûts de développement du véhicule sont d'environ 20 000$.

References

[1] Killough S.M., Pin F.G., "Design of an Omnidirectional and Holonomic Wheeled Platform Prototype", *Proceedings of the 1992 IEEE International Conference on Robotics and Automation*, Nice, pp.84-90.

[2] Giralt G., Chatila R., Vaisset M., "An integrated Navigation and Motion Control System for Autonomous Multisensory Mobile Robots", Autonomous Robot Vehicles edited by Cox I.J., Wilfong G.T., Springer-Verlag, pp.420-443.

[3] Blais F., Rioux M., Domey J., "Optical Range Image Acqusition for the Navigation Of a Mobile Robot", *Proceedings of the 1991 IEEE International Conference on Robotics and Automation*, Sacramento, pp.2574-2580.

[4] De Santis R.M., Hurteau R., Piché A., "On the Motion Control of Electric AGVS" *Second Workshop on Military Robotic Applications* Kingston, August 1989, pp 306-315.

[5] Latombe J.C., "Robot Motion Planning", Kluwer Academic Publisher, 1991.

[6] De Santis R.M., "Path-Tracking for Car-Like and Tractor-Trailer-Like Robots", Rapport technique, École Polytechnique, EPM/RT-93/25, Nov. 1993.

[7] Juneau L., Hurteau R., Freedman P., Chevrette G., "Using Laser Range Data to Model Tunnel Curvature for the Automatic Guidance of a Mining Vehicle", Second IEEE Conference on Control Applications, Sept 1993, Vancouver, pp.643-648.

90C 214

ANCAEUS Navigation Module

Terrence J. Nelson, M.Sc, B.Sc., B.E.
APRO Applied Robotics Inc., 212-112 Research Drive
PO Box 831, Saskatoon, Saskatchewan, S7K 3L7
Ph: (306) 242-2776, Fax: (306) 242-9767

#149652

Abstract

The ANCAEUS System is a tele-operated or semi-autonomous unmanned vehicle system that has been developed at the Defense Research Establishment (Suffield). The ANCAEUS System was developed as a general purpose system capable of supporting a wide range of military unmanned vehicle applications. The ANCAEUS System has been implemented on several vehicles including a Hazardous Materials Handling Robot and the Remotely Operated Metallic Mine Detection System (ROMMIDS) .

The ANCAEUS Navigation Module provides the system with the position and attitude of the ANCAEUS vehicle. The position is determined by using the satellite based Global Positioning System (GPS), and the roll and pitch are determined by using pendulum sensors. The sensors are interfaced to a microprocessor based system that calculates the position and attitude, and monitors the alarm thresholds. The Navigation Module communicates to the Vehicle Control Processor (VCP) by using the ANCAEUS vehicle's fibre optic network.

The position is used by the ANCAEUS System to provide way-point control of the vehicle in automatic (semi-autonomous) mode, and to update the vehicle's position on the moving map display at the ANCAEUS Ground Station.

This paper will discuss the development and implementation of the ANCAEUS Navigation Module and an overview of the capabilities of the ANCAEUS System.

Introduction

The ANCAEUS System is a general purpose unmanned vehicle system that can be easily adapted to any type of vehicle. A typical ANCAEUS System configuration is given in Fig. 1.

The ANCAEUS Ground Control Station (AGCS) is a general purpose ground control station that can be used to control a number of different vehicles (including different types of vehicles) at the same time. The AGCS provides the operator with live video from the selected vehicle, a moving map display, and overlayed status and control data. The AGCS is portable so it can easily be adapted to operate in a variety of situations.

The Vehicle Control Processor (VCP) is the main subsystem for the unmanned vehicle. It acts as a bridge between the Radio Frequency (RF) link and the Fibre Link network on the vehicle. The VCP provides most of the functionality common to unmanned vehicles and also provides extra processing power to run application specific software.

The ANCAEUS Personality Module provides the system with the vehicle specific functions. The Personality Module is responsible for translating vehicle specific commands like 'Start Engine' into actions. The Personality Module concentrates the vehicle specific commands so that the other modules are independent of the vehicle. This allows the ANCAEUS System to be easily implemented on a new vehicle because the Personality Module is the only Module that should have to be redesigned.

The ANCAEUS Navigation Module provides the system with the unmanned vehicles position, orientation, and attitude.

This system is completely independent from the vehicle and can be easily mounted on any ANCAEUS vehicle. This paper will discuss the design and implementation of the ANCAEUS Navigation Module. The position is determined by a GPS satellite receiver and the attitude is monitored by using roll and pitch sensors. The module interfaces to the ANCAEUS Vehicle Control System and provides the information to the ANCAEUS System.

Interface Specifications

The ANCAEUS Navigation Module has 3 interfaces: Power Interface, Fibre Interface, and the GPS Antenna.

The Power Interface (J1) receives a +24 VDC input and daisy chains the +24 VDC to the next module. The Navigation Module can be powered by DC voltages ranging from +12 VDC to +28 VDC.

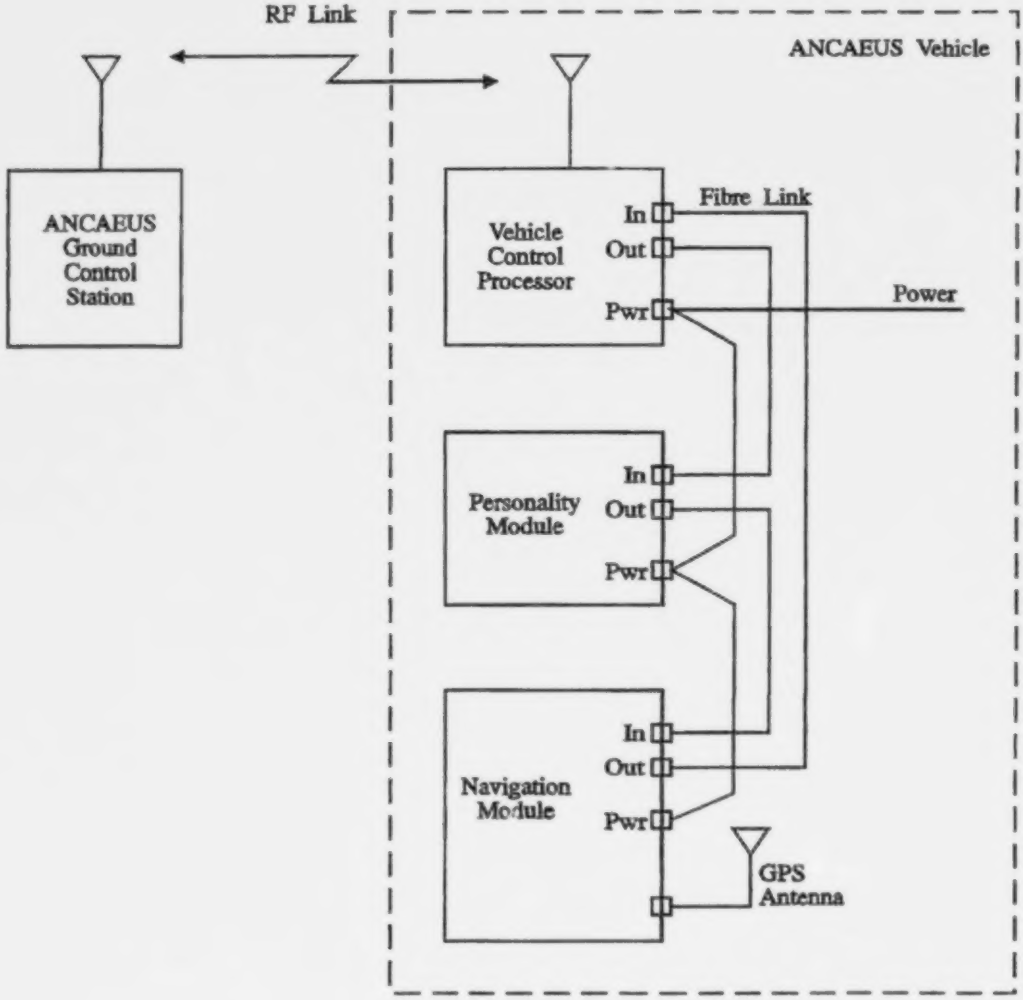

RF Link
ANCAEUS Vehicle
ANCAEUS Ground Control Station
Vehicle Control Processor
In
Out
Pwr
Fibre Link
Power
Personality Module
In
Out
Pwr
Navigation Module
In
Out
Pwr
GPS Antenna

FIGURE 1: ANCAEUS System Configuration.

The Fibre Interface is a fibre-optical communications interface that provides the Navigation Module with the ability to communicate to other modules on the vehicle. The Fibre Interface is a daisy chained fibre optic-link that connects the modules together on the ANCAEUS vehicle. There are two fibre-optical cable connectors: FibreIn (J2), and FibreOut (J3). The FibreIn port receives messages from the fibre-optic network. The FibreOut port transmits messages to the next module. The Fibre Optical Network transmits HDLC messages to all the nodes in the network.

The GPS antenna port (J4) connects the Navigation Module to an externally mounted GPS antenna.

Functional Design

The functional design block diagram for the Navigation Module is given in Fig. 2. Initialize Navigation Module (InitNavModule) is a software

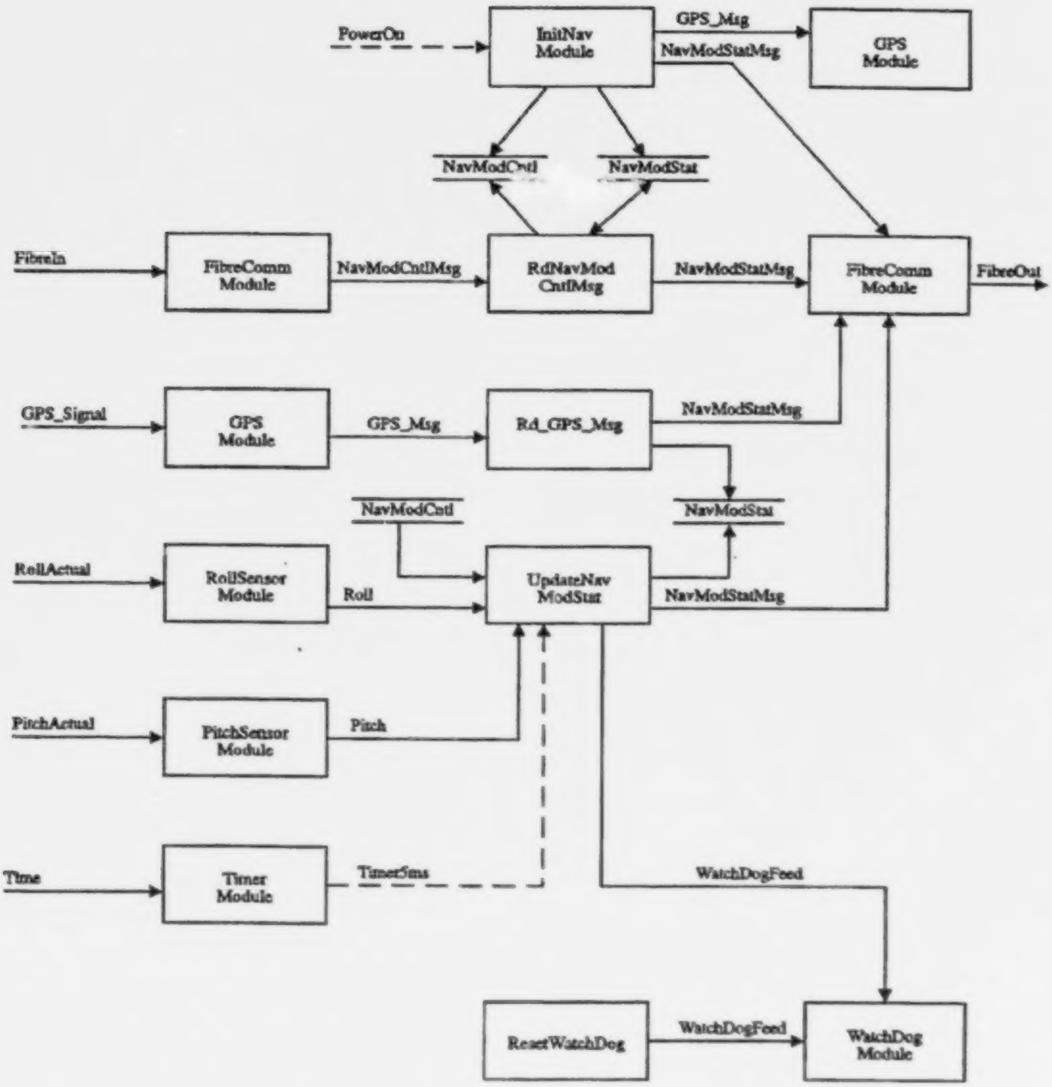

PowerOn
InitNav Module
GPS_Msg
NavModStatMsg
GPS Module
NavModCntl
NavModStat
FibreIn
FibreComm Module
NavModCntlMsg
RdNavMod CntlMsg
NavModStatMsg
FibreComm Module
FibreOut
GPS_Signal
GPS Module
GPS_Msg
Rd_GPS_Msg
NavModStatMsg
NavModCntl
NavModStat
RollActual
RollSensor Module
Roll
UpdateNav ModStat
NavModStatMsg
PitchActual
PitchSensor Module
Pitch
Time
Timer Module
Timer5ms
WatchDogFeed
ResetWatchDog
WatchDogFeed
WatchDog Module

FIGURE 2: Navigation Module Functional Block Diagram

module that initializes the Navigation Module when the power is applied. The Fibre Communications Module (FibreCommModule) provides the hardware and software to interface the Navigation Module to the Fibre-optics communication network. The GPS Module is a firmware module that provides the hardware and software to interface the Navigation Module to the GPS signal from the satellite. The Roll Sensor Module and the Pitch Sensor Module provide the hardware and software to monitor the attitude. The Timer Module provides the real-time control for the Navigation Module. The Reset Watch Dog and the Watch Dog Module monitors the operation of the microprocessor. Read Navigation Module Control Message (RdNavModCntlMsg) is a software module that reads messages from the Fibre Communications Module (FibreCommModule) and processes the commands. The Do_GPS_Module receives messages from the GPS Module and translates them into position and orientation. Update Navigation

Module Status (UpdateNavModStat) monitors the Navigation Module and maintains the current status and alarm conditions.

Implementation

The ANCAEUS Navigation Module was implemented as shown in Fig. 3 The Roll Sensor and the Pitch Sensor are Humphrey Model No. CP17-0622-1 Pendulum Sensors that can be mounted in the Navigation Module in the required orthogonal plane to provide roll and pitch attitude information. The GPS Engine is a Trimble Model SV6 capable of single ended or differential GPS operation. The Fibre I/O Printed Circuit Board Assembly is a small printed circuit board that is common in all ANCAEUS modules to interface the Fibre network to the module. The NavigatioModule Printed Circuit Board Assembly is a microprocessor based circuit board that provides all of the functionality of the Navigation Module and interfaces the sensors to the ANCAEUS System.

Conclusion

The ANCAEUS Navigation Module was successfully implemented and tested in the ROMMIDS in the spring of 1993. Since that time it has proven itself to be a reliable system to provide position and attitude information for the ANCAEUS unmanned vehicles.

Applied Robotics is currently investigating the commercial application of the Navigation Module and the ANCAEUS System to a number of other sectors including mining and agriculture.

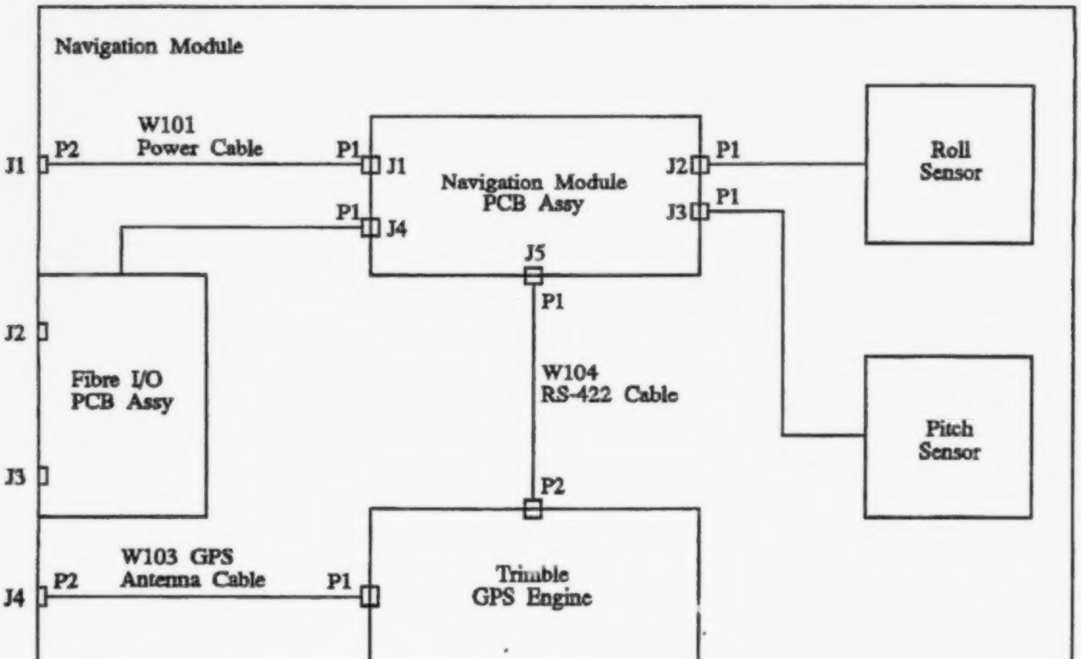

Navigation Module
W101
Power Cable
J1
P2
P1
J1
Navigation Module
PCB Assy
J2
P1
J3
P1
Roll
Sensor
P1
J4
J5
P1
J2
Fibre I/O
PCB Assy
W104
RS-422 Cable
Pitch
Sensor
J3
P2
W103 GPS
Antenna Cable
J4
P2
P1
Trimble
GPS Engine

FIGURE 3: Navigation Module Cabling Diagram

904714

Dead Reckoning for Mobile Robots via Sensor Fusion

D.N. Green & J.Z. Sasiadek
Department of Mechanical and Aerospace Engineering
Carleton University
Ottawa, Ontario, Canada K1S 5B6
Telephone: (613) 788-2600 x5698 FAX: (613) 788-5715

D.S. Necsulescu & B. Kim
Department of Mechanical Engineering
University of Ottawa
Ottawa, Ontario, Canada K1N 6N5
Telephone: (613) 564-5761 FAX: (613) 564-5879

#149654

Abstract

This paper presents a technique and practical algorithm for the combining or "fusion" of signals from odometry and an Inertial Navigation System (INS). This fusion mitigates the problems encountered when using either system alone, including wheel slippage when using odometry and integration-error when using an INS. An experimental vehicle, fitted with a wheel-encoder and an accelerometer, is used to acquire data for testing of the system. The signal processing and fusion algorithms are described and tested using the acquired experimental data. The resulting signal fusion is shown to reduce the effects of both wheel-slip and integration error, resulting in more reasonable estimates of position and velocity than are observed using odometry or the INS alone.

I. Introduction

Errors in odometry usually result from changes or inaccuracies in wheel radii, wheel slippage and deformation of the ground surface. Indoor mobile systems seem to suffer the least from these wheel/ground interaction phenomena, and linear position accuracies of ±0.5% are not uncommon [1,2]. In contrast to indoor systems, an autonomous off-road vehicle usually suffers from all of the above problems. Furthermore, irregular changes in wheel/ground interaction make it difficult to determine and compensate for measurement errors.

The navigation of an autonomous planetary rover is of particular interest in this work, and it has been suggested that an Inertial Navigation System (INS) might suit the navigation needs of such a vehicle and its mission [3]. Such a system would avoid problems such as wheel slippage, and should be able to maintain a position accuracy similar to that of an indoor odometry system. There are, however, still a number of benefits to using odometry as opposed to an INS technique, including: high physical robustness and reliability of the encoders, low weight, power consumption and cost, and a zero time-dependent drift. This last point serves to point-out that INS errors (in position and velocity) increase over time, regardless of how fast or far the vehicle travels.

It was proposed earlier that odometry and inertial navigation might be combined somehow, in an effort to mitigate the problems encountered when using either system alone [4]. In this work, the signals from a wheel encoder (oedometer) and a linear accelerometer (INS) have been combined or "fused", in an extension of that earlier work. Section II, below, describes the experimental apparatus and the acquired data. Section III describes the primary signal processing operations, and Section IV develops and describes the fusion process, through analysis of the experimental data.

II. Experimental Data

The experimental vehicle consisted of a three-wheel mobile robot with a single forward wheel for propulsion and steering. An optical position encoder was used to measure the rotational (and inferred linear) displacement of the drive wheel. A linear accelerometer was mounted near the drive wheel to measure the forward acceleration of the vehicle. The signals from these two sensors were sampled simultaneously at a rate of 100Hz. The resulting discrete data streams, referred to simply as the encoder and INS measurement signals, were then used in the development and testing of the data fusion technique.

A counter-weight was extended from the rear of the vehicle to lower the drive traction and encourage wheel slippage. The position controller was then commanded to drive the robot in a straight line, to a position 2m from the starting point, while using the encoder signal for feedback. Figures 1-3 illustrate results from a typical test run, where signals other than the encoder position and INS acceleration were obtained though differentiation and integration, as will be described in Section III.

In Figure 1 it can be seen that, according to the encoder signal, the vehicle overshoots and reverses to settle at a position just short of the 2m set-point. (The steady-state error is due to internal mechanical friction and play.) The INS signal (Figure 1), however, indicates that the vehicle position increases more slowly, overshooting at a somewhat lower position, and then continuing to increase without apparent bound. These differences are explained with reference to the velocity responses.

In Figure 2, the encoder velocity signal can be seen to increase and then oscillate above the INS signal, in the regions of about 1.5s to 3.5s and 4.5s to 5.5s. This phenomenon was caused by slip-induced oscillation in the feedback-control system. The INS signal, although not subject to the effects of wheel slip, suffered from the problem of integration error. After about 5.5s, the rover was observed to have come to a complete stop, while the INS velocity signal indicates that the rover continued to move forward at a significant speed.

In summary: When no drive-wheel slippage is observed, the encoder-based estimates appear to be valid, but when significant slip occurs, the encoder tends to overestimate the change in position, which also yields an erroneous velocity estimate. On the other hand, although the INS-based estimates suffer from an increasing error in integration, the INS system is itself free from the effects of wheel slip. The complementary nature of the observed characteristics suggests that fusion of the encoder and INS signals will result in an overall improvement in the estimates of position and velocity.

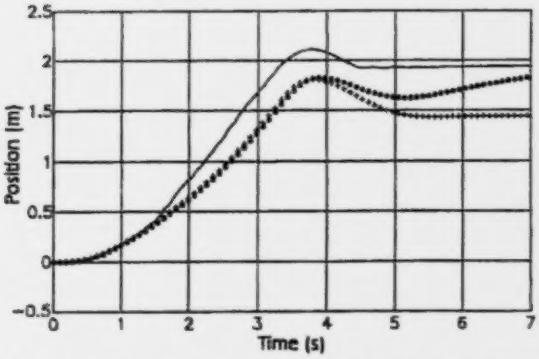


Figure 1: Encoder (—), INS (o) and fused (+) position.

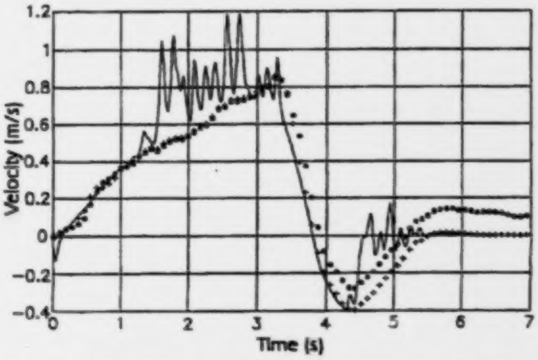


Figure 2: Encoder (—), INS (o) and fused (+) velocity.

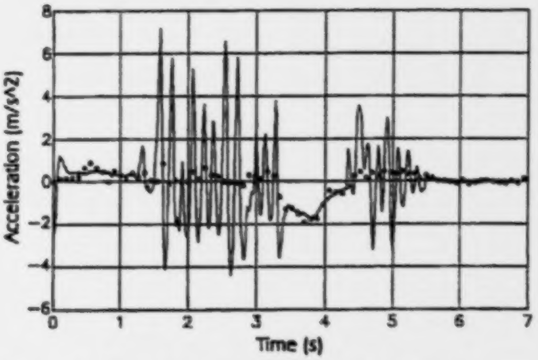


Figure 3: Encoder (—) and INS (o) acceleration.

III. Signal Processing

Consider a system (vehicle motion) that is described by a discrete-time model and the following state-space equations:

$$x(k+1) = Ax(k) + Bu(k) + B_1 w_1(k)$$

$$y(k) = Cx(k) + w_2(k) \tag{1}$$

$$y_T(k) = Cx(k)$$

where, at sample moment k, $x(k)$, $u(k)$, $y(k)$ and $y_T(k)$

are the state, input, sensor output and true output vectors respectively, $w_1(k)$ is the process disturbance noise, $w_2(k)$ is the measurement noise and A, B, B_1 and C are parameter matrices describing the system. The true output, $y_T(k)$, represents the actual vehicle response that we are trying to determine.

The expressions in (1) may be formulated such that the state vector, x(k), is composed of all relevant states of a system of interest, such as position, velocity and acceleration. In this work, however, these expressions will be used to develop scalar linear filters for various data streams. The state vector, x(k), will in fact be composed of only a single state variable in each application. At each moment k, the sensor output or measurement vector, y(k), is therefore equal to the sum of the true value of the scalar, Cx(k), and a random noise error, $w_2(k)$. (Note that parameter C is used to account for the presence of a measurement or sensor gain.)

The noise vectors $w_1(k)$ and $w_2(k)$ can be defined as random, white-noise sequences, with zero means, and magnitudes described by the expected values of covariance, given by,

$$E\left\{w_1(k)\,w_1^T(k)\right\} = Q, \quad E\left\{w_2(k)\,w_2^T(k)\right\} = R \tag{2}$$

where Q and R are diagonal matrices of expected noise variances for respective elements of the noise vectors [5,6].

The Linear Quadratic Estimator (LQE), is a linear-system state estimator that uses the minimization of the mean-square estimation error as an optimality criterion [5,6],

$$E\left\{\left[\hat{y}(k) - y_T(k)\right]^2\right\} = \min \tag{3}$$

where $\hat{y}(k)$ is the filtered signal or sensor estimate, $y_T(k)$ is the true signal mentioned above, and the expected value of the square-difference between these signals is to be minimized.

The discrete LQE can be described by [5,6],

$$\begin{aligned} \hat{x}(k) &= \bar{x}(k) + L\left[y(k) - C\bar{x}(k)\right] \\ \bar{x}(k) &= A\hat{x}(k-1) \\ \hat{y}(k) &= C\hat{x}(k) \end{aligned} \tag{4}$$

where $\hat{x}(k)$ is the current estimate of the state vector, $\bar{x}(k)$ is the predicted state based on the model and previous estimate, and L is a matrix of gains. The LQE design procedure utilizes a performance criterion that satisfies the minimization in (3), by considering the effects of the known (often estimated) noise matrices, Q and R, in (2). The resulting *optimal* estimator is often referred to as a Kalman filter.

The basic principle of operation of (4) is simply that: with smaller gains (L), the estimate, $\hat{x}$, will be based more on the model prediction, $\bar{x}$, whereas with larger gains the estimate will be based more on the sensor measurements, y. In this work, a first order filter design was chosen to simplify the development of the fusion algorithm. For example, the following operations facilitate the filtering of a sequential stream of velocity measurements:

$$\begin{aligned} \hat{v}(k) &= \bar{v}(k) + L\left[v_m(k) - \bar{v}(k)\right] \\ \bar{v}(k) &= A\hat{v}(k-1) \end{aligned} \tag{5}$$

where $v_m(k)$ is the current velocity sample and $\hat{v}(k)$ is the corresponding filter estimate. The filter parameters, A and L, are scalars in this case, where A is based on the system characteristics, and L must be chosen to yield a satisfactory filter response. For the vehicle model in this work, the parameters were set to $A=0.995$ and $L=0.3$, where L was initially chosen according to (2) and (3), and later adjusted to this value, through experiment, to yield satisfactory results.

The filter in (5) was developed initially to remove sharper spikes from velocity data, however it was also found to be useful for "smoothing" at other points in the signal path. The filter locations can be seen in the signal flow-chart in Figure 4, where each serves to smooth recently processed signals and/or prepare signals for further processing (usually differentiation). The ENCxxx and INSxxx terms, in the chart, refer to processed encoder and INS-based signals, respectively. Samples up to and including the current sample, k, are kept available for external processing. Signals ε and $\dot{\varepsilon}$, referred to as acceleration error and the acceleration error rate, respectively, are used by the fusion algorithm discussed next.

IV. Signal Fusion

It was pointed out in Section II, that when there is little or no wheel-slip, encoder estimates of position and velocity should generally be more accurate than the "drifting" estimates obtained through integration of the INS signal. On the other hand, at the onset of wheel-slip, the INS signals should maintain a much more reliable baseline from which to estimate changes in the true position and velocity of the vehicle. The problem then, is to determine the onset and presence of wheel slippage, and to update the position and velocity estimates using the appropriate signal source.

At the acceleration level (Figure 3), neither the encoder nor the INS signals suffer from an accumulating (integration) error, so it is at this level that we can reliably detect the onset of wheel slippage. In Figure 3 one can see that during the periods of about 1.5s to 3.5s and 4.5s to 5.5s, there appears to be considerable difference between the two signals, where the large changes in the encoder signal are assumed to indicate the presence of wheel slippage, as mentioned earlier. Figure 5 contains a plot of the difference between the acceleration signals, as well as a plot of the rate of change of that difference. The determination of these two new signals, referred to as ε and $\dot{\varepsilon}$, is depicted in the chart in Figure 4. These new signals are 90° out of phase from each other, and throughout the wheel-slip intervals mentioned above, one or both of theses new "error" signals will have a relatively large magnitude. Observation of the signal magnitudes, $|\varepsilon|$ and $|\dot{\varepsilon}|$, should facilitate detection of significant wheel slippage.

In addition to differentiation, integration and filter subroutines, a C-langauge routine was written to complete the signal fusion process. The following code describes the subroutine:

```
if:
    $|\varepsilon| \le \varepsilon_{max}$   and   $|\dot{\varepsilon}| \le \dot{\varepsilon}_{max}$
then:
    v(k) = ENCvel(k)                                      (6)
    p(k) = p(k-1) + ENCpos(k) - ENCpos(k-1)
else:
    v(k) = v(k-1) + INSvel(k) - INSvel(k-1)
    p(k) = p(k-1) + INSpos(k) - INSpos(k-1)
```

where ε_{max} and $\dot{\varepsilon}_{max}$ are maximum error set-points, the ENCxxx and INSxxx terms represent signals taken from correspondingly labelled points in the signal paths in Figure 4, and v(k) and p(k) are the current fusion estimates of velocity and position, respectively. The operation of this routine is as follows: If the magnitude of the acceleration error (ε) and error rate ($\dot{\varepsilon}$) signals are each within their respective maximum set-points, then the velocity and position estimates are updated using encoder data. If either error signal is greater than its set maximum, then the estimates are updated with INS data. The results of the fusion of the encoder and INS data streams are illustrated in the position and velocity plots contained in Figures 1 and 2. In these examples, the error maximums were set to $\varepsilon_{max} = 0.25\text{m/s}^2$ and $\dot{\varepsilon}_{max} = 2.5\text{m/s}^3$.

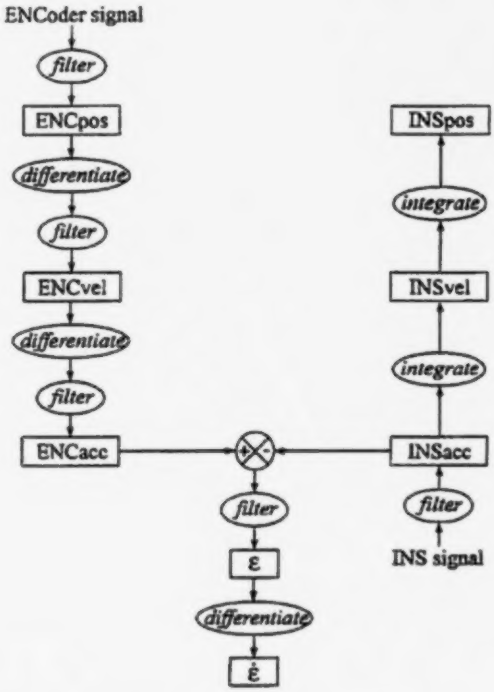


Figure 4: Encoder and INS signal paths.

During about the first 1.5s of motion, the encoder, INS and fused signals follow each other fairly closely. (It is noted that there is an obvious error in the early encoder velocity signal, which results in the fusion estimate following the INS signal very closely.) At about 1.5s, the drive wheel begins a series of slip-related oscillations, which continue for about 2s. During this slipping motion, the fused signal can be

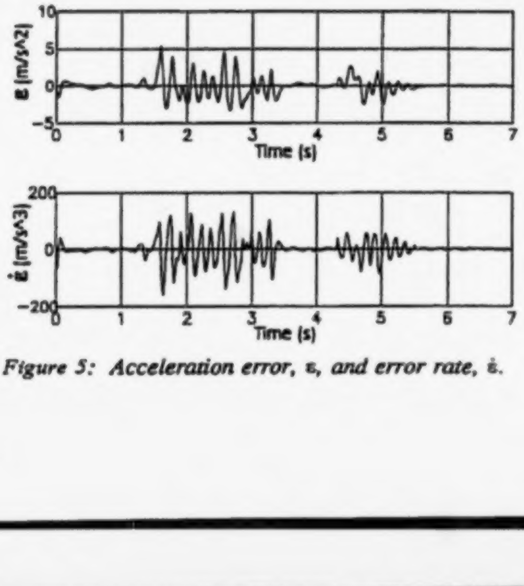


Figure 5: Acceleration error, ε, and error rate, $\dot{\varepsilon}$.

seen to follow the INS signal rather closely. At about 3.5s into the motion, the drive wheel discontinues slipping, and the fused velocity signal can be seen to "migrate" towards the encoder signal, while the fused position estimate can be seen to begin paralleling the encoder signal. It should be noted that the encoder-based fusion estimate of velocity, v(k) = ENCvel(k) in (6), is actually passed through an additional Kalman filter, to avoid sudden jumps in the fused signal. At about 4.5s, the drive wheel begins slipping once again, and the fused velocity signal begins to parallel the INS signal. At this point it should be noted that the INS velocity signal has actually drifted noticeably in the positive direction, but that the fused estimate is based only on relative changes in the INS signal.

After about 5.5s the drive wheel has stopped slipping and the vehicle has come to a stop. The encoder and INS acceleration signals continue near zero, and the fused signal continues to update using the encoder signal. The INS position does not settle due to increasing integration error, however the fusion estimate of the final position is observed to remain constant, at considerably less than the slip-induced over-estimate by the encoder.

V. Additional Comments

In this work, a rather conservative approach has been taken to the design of the fusion algorithm and the choice of signal sources. It is noted that the use of a tachometer, for odometry, would yield a signal that is at neither the acceleration nor position "level". The use of such a rate signal would likely improve the performance of the fusion algorithm, by reducing the required amount of signal processing. In this work, however, inertial navigation is represented only by acceleration, and odometry by displacement. It is believed that this greater disparity between odometry and INS signal "levels" may lead to the development of a more complete (and perhaps more robust) fusion technique. From a practical standpoint, the available test vehicle was not configured for use with a tachometer upon initiation of this work.

In light of the above-mentioned benefit to reducing the amount of signal processing, it should be further emphasized that the differentiation processes, indicated in Figure 4, do indeed add noise to the fused position and velocity estimates. During experiment, the position of the test vehicle was closed-loop controlled using the wheel encoder signal for feedback, as mentioned earlier. It is intended, of course, that in future applications, the control loop should be closed by feedback of the fused estimates, and not by the original signals. The stability of such a fused-signal-feedback system will decrease with increasing noise in the fused estimates. While the optimum configuration and characteristics for the fusion process have not yet been determined, it is clear that an effort must be made to minimize the amount of signal processing.

VI. Conclusion

A technique has been developed and shown to be capable of fusing odometry and INS signals into a single set of velocity and position estimates. The fused signal was affected very little by induced wheel slippage in the experiment, and suffered only minimally from the integration errors associated with an INS. The fusion procedure is straight-forward, and the algebraic algorithms are simple enough for real-time applications. Although there are numerous terrain/vehicle scenarios to be considered, the technique applied here is an encouraging point from which to continue the investigation of signal fusion for autonomous vehicle navigation.

References

[1] J. Borenstein and Y. Koren, "Motion Control Analysis of a Mobile Robot," *J. Dyn. Sys., Meas.,Cntrl.*, vol. 109, pp. 73-79, June 1987.

[2] J. Holland, H.R. Everett and G.A. Gilbreath, "Hybrid Navigational Control Scheme for Autonomous Platforms", *Mobile Robots V*, Proc. SPIE, vol. 1388, pp. 291-298, 1991.

[3] D.P Miller, A.H. Mishkin, K.E. Lambert, D. Bickler and D.P. Bernard, "Autonomous Navigation and Mobility for a Planetary Rover," *27th Aerospace. Sciences Meeting*, AIAA, #89-0859, 1989.

[4] D.S. Necsulescu, J.Z. Sasiadek, B. Kim and D.N. Green, "Fusion of Kinematic and Inertial Navigation Systems for Autonomous Vehicles", *Proc. IEE-IEEE Vehicle Navigation & Information Systems Conference - VNIS '93*, pp. 462-465, Ottawa, October 1993.

[5] J.Z. Sasiadek and P.J. Wojcik, "Design of a Discrete Kalman Filter for a Tactile Sensor in a Feedback Control System", *Intl. Journal of Control*, vol. 45, no. 4, pp.1211-1228, 1987.

[6] G.F. Franklin, J.D. Powell and M.L. Workman, *Digital Control of Dynamic Systems*, Second Edition, Addison-Wesley, 1990.

904077

Electrochemical Power Sources for Robotic Systems

A. V. Anantaraman and W. A. Adams
Electrochemical Science and Technology Centre, University of Ottawa
Ottawa, Ontario, K1N 6N5

#149656

Abstract

Robotic technology has made a significant impact in tactical military applications and hazardous environments. Commercial and scientific sectors have long been aware of the potential uses of Autonomous Underwater Vehicles (AUV) and there are a number of established programmes aimed at developing such vehicles. A key component in the design and development of AUVs and robotic systems in general, is the power source which should be carefully integrated early in the design phase.

This paper examines advanced electrochemical power sources that can perform reliably in long range/long duration AUV missions. The complex design process of AUVs is analyzed in terms of specific parameters. A mathematical model based on propulsion hydrodynamics is used to assess the interactive coupling between the power source system and the mission parameters.

Introduction

It is generally recognized that war has been one of mankind's most passionate and enduring endeavours. Apart from the hazards of modern combat environment, several factors favour the adoption of robotic technology for military applications. For instance, the possible use of nuclear, biological and chemical threats, the demand for highly skilled military manpower capable of operating complex integrated weapon systems, and the excessive physical constraints on the individual soldier, have made a strong case for the inevitable use of robotics in all the three forces. The strategy outlined by the Soldier Modernization Plan (SMP) views the soldier as a system to enhance the functions that have a direct impact on the soldier effectiveness: lethality, command and control, protection, sustainment and mobility. Robotic vehicles, both aerial and underwater, and the SMP, are examples of new defense strategies that will continue to demand increasing roles for this technology.

The Autonomous Underwater Vehicle (AUV) has found a wide range of acceptance both in the military and civilian applications. AUVs are being used in underwater surveillance, anti-submarine and anti-surface warfare, covert deployment of long range stand-off weapons and counter measures. Commercially AUVs are being used increasingly in off-shore explorations and deep sea salvage operations of sunken submarines and ships. AUVs are used in scientific studies of ocean floor survey and mapping.

While research and development efforts in robotics have kept pace with the related areas of computer technology and material science, there are relatively few studies devoted to power sources which are vital to the operation of the robotic system. This paper examines the various electrochemical power sources available for AUVs and calculates the power and energy requirements for specific missions.

AUV Configuration and Design Parameters

In view of the growing interest in the military and commercial sectors, AUVs were chosen for modelling studies for power source design. It must be pointed out that military applications impose severe constraints on the choice of electrochemical power sources. For example, AUVs for defensive or offensive manoeuvres, will have a power source that meets the following requirements:

(a) extended range and manoeuvrability
(b) silent operation
(c) minimum weight and volume

(d) low temperature operation to avoid IR signature
(e) ability to switch to high power output
(f) low iron content to avoid electromagnetic detection

Selection of power sources for commercial AUVs are less restrictive. For this reason, this study is non-specific and only the basic modelling parameters applicable to the general class of AUVs are considered.

The Autonomous Underwater Vehicle is defined as an unmanned, untethered, underwater vessel that can operate independently. At present, two different classes of vehicles are identified in accordance with its functions necessary to accomplish the generic missions. One is a long-range autonomous vehicle and the other is a short-range/short duration vehicle. A typical mission for a long-range vehicle involves military surveillance or ocean survey for several hours; a short duration mission would be a local area search and observations.

Over the past 20 years, about 45 AUVs have been built worldwide, half of which comes from the United States of America. Notable among these are the Advanced Underwater Search System (AUSS) and an experimental vehicle, EAVE EAST, used for pipeline and platform inspection. Canadian efforts in AUV development have progressed on two fronts: deep water submersibles and near surface AUVs. The DOLPHIN (Deep Ocean Logging Profiler with Hydrographic Instrumentation and Navigation) is an 8m long, 2500kg, 15 knot, radio remote controlled and reporting, mini snorkeling submarine. It has a demonstrated capacity of an offshore survey vehicle (1). The second vehicle called ARCS (Autonomous Remote controlled Submersible) is a 7m long, 200kg, 5 knot, battery powered vehicle that can be used for under ice survey, surveillance and reconnaissance. A smaller version of this vehicle has also been developed (1,2).

The basic components of an AUV are the following:

(a) Payload, control and data acquisition and storage, navigation and communication systems. This set is generally referred to as payload for convenience;

(b) Main energy system;

(c) Emergency energy system;

(d) Power conditioning and distribution, propulsion, steering and stabilizer systems;

(e) Buoyancy foam (ballast) for weight-buoyancy balance;

While the total mass and displacement must include the contributions from each of the above, the primary energy source and its ancillary subsystems represent the bulk of the payload.

In a recent analysis of the parameters that are considered important in the design of an AUV, Brighenti (3) classifies them in two main sets:

Mission related parameters:
(a) payload
(b) range or endurance
(c) maximum operating depth
(d) average cruise speed
(e) maximum cruise speed

Design related parameters:
(a) type of power generation system
(b) type of energy storage system
(c) type of reactants to be stored
(d) type of storage vessels
(e) type of ballast used for balancing the vehicle

It is important to point out that the mission parameters must be fixed first; this leads to approximate values of energy requirements which can be used to formulate and update the other design parameters.

Batteries and Fuel Cells

For comparison of electrochemical power systems, it is usual to refer to the battery capacity in terms of two parameters:

$$Specific\ energy\left(\frac{Wh}{kg}\right) = \frac{Delivered\ Energy}{Battery\ Mass}$$

$$Specific\ power\left(\frac{W}{kg}\right) = \frac{Rated\ Available\ Power}{Battery\ Mass}$$

It is sometimes found useful to define the specific energy and specific power in terms of battery volume. When making a comparative assessment of different batteries, it should be realized that the amount of energy available from a cell or battery depends on how quickly it is drawn; in other words, the energy density decreases as the power density is increased. Hence when comparing the energy densities of different batteries, it is necessary to define the rate at which the energy is withdrawn.

Many of the AUVs in operation throughout the world today are battery powered. Secondary lead-acid batteries are by far the most commonly used, because of their reliability, low cost and relative safety. Nickel-cadmium and silver-zinc batteries have been used in a few vehicles. Advanced high energy density batteries and fuel cells are now being designed specifically for AUV applications (4,5). For example, Marconi's prototype AUV system has been undergoing trial tests using sodium-sulphur batteries. These high temperature batteries operate at 295-300°C. The mean cell voltage up to 40% discharge is about 2.04 V. The coulombic efficiency of the battery is 100%, because the electrolyte beta-alumina is perm-selective to only sodium ions, thereby eliminating secondary reactions. Non-magnetic lithium cells are also considered for use in AUVs (6). Lithium based batteries have a wide operating temperature range, particularly at low temperature and provide high energy and high power densities even at high discharge rates. A comparative assessment of various advanced batteries for military tactical purposes are available in reports (7,8). Data on recent AUV systems are shown in Table I (9).

TABLE I: AUV Data and Power Sources

Year	Vehicle	Mission	Depth (m)	Dimensions (lxwxh)cm	Speed (knots)	Power source	Weight (kg)
1979	Eave East	Test-bed	91	129 x 135 x 135	1.5	Pb-acid	272
	Eave West	Test-bed	670	270 x 56 x 56	1.8	Pb-acid	
1983	AUSS	Search	6096	247 x 78 (diam)	7.0	Ag-Zn	907
	Epaulard	Topogr'y	6000	400 x 110 x 200	2.0	Pb-acid	
1984	ARCS	Under-ice	400	70 x 70 (diam)	6.0	Pb-acid	1510
1986		R&D				Pb-acid	
1987	LSV	Submarine testing	n/a				
1988	MUST	Test-bed	610				
1989	MT-88 (Russia)	Sea bottom survey	6000				
1990	UROV-2000 (Japan)	Bottom survey	2000				
1992	Dolphin (Glasgow, Scotland)	Ocean monitoring	6000				
1993	ARUS (European)	Bottom survey	n/a				

Fuel cells are batteries in which the chemical energy of a fuel and oxidant is converted into low voltage direct current by the electrochemical reactions that take place at the anode and cathode. During normal operation, the reactants are continuously fed to the electrodes and the product is continuously removed. In analogy, a battery may be conceived as a fuel cell where the reactants and products are internally stored. In the hydrogen/oxygen fuel cell, the reactants, hydrogen and oxygen, are continuously supplied to the electrodes from an external storage and the product water is removed.

The Proton Exchange Membrane (PEM) fuel cell uses a thin ion exchange polymer film as an electrolyte. The operating principle of the fuel cell is shown in Fig. 1.

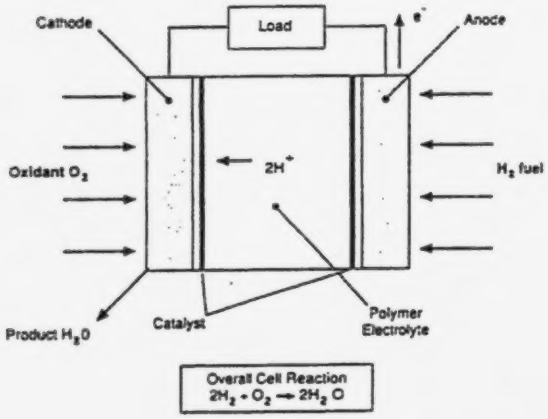


FIGURE 1: PEM Fuel Cell

Different types of fuel cells are identified depending on the electrolyte and the operating temperature. Table II summarizes the types of fuel cells that are being actively developed for various applications.

Although there are many fuel-oxidant combinations that can be used in a fuel cell, for practical and economic reasons, the hydrogen-oxygen fuel cell system is the preferred choice. Because of the risk associated with the storage of pure hydrogen, hydrogen rich compounds such as methyl alcohol and natural gas are used as sources of hydrogen.

Along with electrical power, fuel cells produce heat and water. Management of the products, heat and water, and the containment of electrolyte, are technical and engineering problems that have challenged the development of fuel cells. However the state-of-the-art in fuel cells has evolved significantly in the past decade. In particular, PEMFC has progressed to a point where major commercial and military applications are under way. In addition to the general advantages of fuel cells over rechargeable batteries, PEMFC has inherent features that make it a prime candidate for use in closed environments. The main advantages of this fuel cell are:

(a) The electrolyte being a solid, is invariant and stationary.
(b) The only liquid in the fuel cell is water, so that the system suffers minimum corrosion.
(c) It requires minimum volume and has shown good reliability and performance at low temperatures.

Criteria for Power Sources for AUVs

The choice of a suitable electrochemical power source for an AUV is a complex trade-off among the many design variables and mission requirements. The mission parameters such as the nature and duration of the mission, are major factors in the selection of an optimum power source. The propulsion system of an autonomous underwater submersible consists of a set of interdependent subsystems, as shown in Fig. 2.

For most applications, the power demand varies in relation to load. Hence the power source design must be based on the maximum power demand of the

TABLE II: Fuel Cells and their Characteristics

Type	Temperature °C	Electrolyte
Alkaline (AFC)	60 - 80	KOH
Phosphoric Acid (PAFC)	180 - 220	Phosphoric Acid
Polymer (PEMFC)	80	Proton conducting membrane
Molten carbonate (MCFC)	650	$(Li_2CO_3 - K_2CO_3/LiAlO_2$
Solid oxide (SOFC)	1000	$Y_2O_3 + ZrO_2$

vehicle. For batteries, it is usual to compare the specific energy and specific power of individual systems, from known data. The actual available energy depends on operational variables, such as rate of discharge and temperature. Because of space considerations in AUVs, it is necessary to consider both the mass and volume ratios of the energy and power densities.

In order to assess the potential of different electrochemical power systems to meet the needs of an underwater vehicle, operational objectives must be clearly defined. This task is complicated because of the fact that many of the AUVs are experimental test beds or prototypes, and hence technical requirements have not been fully developed.

Power sources for underwater applications must provide high power density and high energy density. The operational practice of maintaining constant buoyancy requires that the power system maintains constant mass and volume for the duration of the mission. Based on these criteria, performance characteristics of advanced batteries and fuel cells that are potential candidates for AUV applications are given in Table III.

For fuel cells, the reactant storage and auxiliary equipment must be included in evaluation of the specific energy. One of the key issues is the storage of hydrogen and new technologies for hydrogen storage and transport are being studied. Use of oxygen from ambient air may need prior treatment if the air environment is contaminated and saturated with salt aerosol.

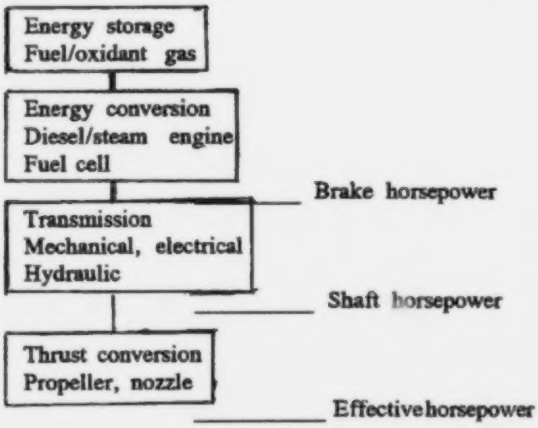


FIGURE 2: Propulsion system

Modelling Studies

In the past, selection of power systems for underwater vehicles has been rather arbitrary, based on known data on performance and power/energy densities. The purpose of this study is to obtain a realistic estimate of the size, weight, and power for a conceptual model of an underwater vehicle. The initial task is to define a set of sub-systems that would be required to accomplish the hypothetical task. Two vehicle shapes have been described, a torpedo shape and a low drag shape. For these shapes the weight and volume occupied by the payload and subsystems, other than propulsion and energy, are the same. This allows the remaining variables, the size and weight o the propulsion and energy subsystems, to be expressed

TABLE III: Power Source Options for AUV Systems (10)

Type	Source	Energy Density	
		Gravimetric Wh/kg	Volumetric Wh/m^3
Primary	Li/SOCl$_2$	412	7.58 x 10^5
Secondary	Lead-acid Ag/Zn Li/Metal S Na/S	35 150 150	8.0 x 10^4 2.0 x 10^5 3.7 x 10^5
Fuel Cell	PEM (Cryo.storage) AFC (Cryo.storage) AFC (Pure gases) PAFC (Cryo.storage) PEM (MeOH/O$_2$)	650 2059 150 580 450	2.0 x 10^5 2.06 x 10^6 1.50 x 10^5 1.80 x 10^5 3.0 x 10^5

as packing densities (in terms of a parameter called prismatic coefficient) and the size and resultant drag that can be achieved with each shape.

Blose and McCartney have developed an iterative algorithm for the design of a propulsion power plant for submarines (11). Hawley and Reader have adopted a similar procedure in the selection of optimum power systems for underwater vessels (12). For the present study, a simple hypothetical mission is used for testing the model. The mission parameters are:

Operating depth: 1000 m
Maximum vehicle speed: 160 m/min
Cruising vehicle speed: 100 m/min
Payload: 800 kg

In an AUV, volume is critical and all subsystems are packaged as densely as possible; on the other hand, mass is of secondary importance as long as buoyancy is within acceptable limits. The design parameters are:

Body shape: torpedo
Vehicle dimensions: 10 m (length) x 1 m (diam.)
Trial power systems: lead-acid battery; PEM fuel cell with cryogenic storage

The vehicle configuration, that is, the packing of the various components within the available space, with minimum expenditure of energy, depends on the mission requirements. Initially, trial values have to be assigned to the parameters, displacement volume of the whole body, wetted surface area, and prismatic coefficient.

The minimum energy requirement for a vehicle moving in a viscous fluid at a given speed can be calculated using known hydrodynamic conditions:

(a) choice of vehicle speed for minimum expenditure of propulsion energy;
(b) choice of body shape for minimum drag;
(c) minimum body envelope to reduce skin friction.

The sum of the two drag terms yields the total drag, which is the force that must be overcome by the propulsion system.

The propulsion power required is proportional to the drag and the velocity and inversely proportional to the propulsion efficiency. The drag experienced by an object moving at a specified speed in a fluid, is an experimentally derived value; hence in the absence of such data, empirical methods are used to approximate the drag. The drag is expressed in terms the surface area and speed

$$D = (1/2)\ P\ S\ V^2\ C_d \qquad (1)$$

where P is the density of the fluid medium, S is the wetted surface area of the vehicle and V if the speed of the vehicle. The total drag coefficient C_d is generally expressed in terms of separable components, C_{dr}, the form drag and C_{df}, the friction drag. The total drag coefficient C_d is given by

$$C_d = C_{dr} + C_{df} \qquad (2)$$

Using eq. (1), the total drag can be calculated. The shaft horsepower (SHP) is then calculated using eq.(3),

$$SHP = (D\ V)/(550\ n) \qquad (3)$$

where n is the propulsion efficiency.

An iteration scheme is implemented in order to achieve an optimum energy requirement for the vehicle. It must be pointed out that the iteration includes buoyancy, payload, energy and power unit because of the coupling between the various individual terms. The rate of change of vehicle shape displacement is used as the criterion for controlling the iteration process. The iteration scheme is shown in Fig. 3.

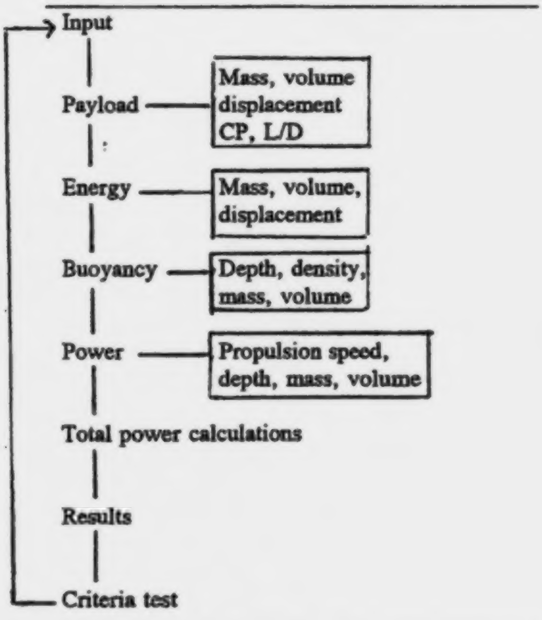

Input
Payload
Mass, volume displacement CP, L/D
Energy
Mass, volume, displacement
Buoyancy
Depth, density, mass, volume
Power
Propulsion speed, depth, mass, volume
Total power calculations
Results
Criteria test

FIGURE 3: Iteration scheme

Discussion

The design of electrochemical power systems for unmanned untethered underwater submersibles is difficult in view of the many unknown parameters and vehicle configurations that are interdependent. The method discussed in this paper provides a simplified basis for examining various power source options prior to the actual design and engineering. The procedure also gives a wider scope for the selection of an optimum power source.

Two different types of electrochemical power systems have been examined, the lead-acid battery and a PEM fuel cell. The program is in the early stages of development and in that respect the results are preliminary and need further improvement.

The PEM fuel cell was chosen because of its many advantages, particularly in the closed environment of a submersible. These include low operating temperature, high performance levels and an excellent electrolyte material that eliminates complicated moving parts. The problems of heat and water management in the fuel cell stack have not been addressed in this study. While this may not be a serious issue in low power fuel cell stacks, it can be a challenging problem in multi-stack assembly because of buoyancy and heat exchanger requirements.

Conclusion

Preliminary results from this study suggest that the method can be used for a systematic first evaluation of various electrochemical power systems. For a given mission duration the power system with the smallest volume has the highest energy density. The lead-acid battery performed poorly in this respect both in terms of mass and volume.

Early results indicate that the PEM provides high power output for long mission duration. The system has a low volume, particularly with cryogenic gas storage. Further work on other advanced power systems are required for a complete evaluation of the model.

References

1. Busby, F. and Vadus, J.R., Autonomous Underwater Vehicles R&D Trends, Sea Technology, May 1990.
2. Yongkuan, L., IEEE Symp. on Autonomous Underwater Technology, 1992.
3. Brighenti, A., IEEE Journal of Oceanic Eng., vol. 15, 179, 1990.

4. O'Leary, J. P., Underwater Technology, vol. 11, 7, 1985.

5. Ruggeri, R. T., 6th International Symp. on Unmanned Untethered Submersible Technology, 153, June 1989.

6. Tonge, A. M. and Cockburn, J., Sea Technology, July 1993.

7. Anantaraman, A. and Roberge, P.R., Power Systems for Military Robotics, DND Workshop, Kingston, Canada, 1989.

8. Christopher, H. A., Gilman, S. and Hamlen, R. P., IEEE AES Magazine, May 1993.

9. Busby, F., Unmanned Underwater Vehicles Data, 1987.

10. MacGregor, J. R., Brown, A. C. and Blane, D. D., Underwater Technology, vol.16, 1990.

11. Blose, J. F. and McCartney, J. F., SAE paper S398, 1964.

12. Potter, I. J., Reader, G. T. and Hawley, J. G., IEEE Symp. on AUV Technology, 1992.

936224

Power System Health Monitoring and Optimization Using Artificial Neural Network

R. G. Cunningham, P. Eng.
Pernix Technology
Dartmouth, Nova Scotia, B2W 4S7

R. D. Mason, P. Eng., Ph D.
Genlink Enterprises Ltd.
Regina, Saskatchewan, S4V 1V2

#149658

Abstract

This paper describes a technology development project done for the Canadian Space Agency under the Atlantic Canada STEAR Initiative, Wave II. The objective of the project was the development of technology useful for power system health monitoring and optimization on solar-powered buoys. A novel aspect of the project is that artificial neural network (ANN) technology is used. The prototype system has been named the Power Monitoring and Optimization System, Buoy variant, or PMOS/B.

Using the Xerion neural network simulator developed by the Connectionist Research Group at the University of Toronto, investigations were carried out on the applicability of ANN solutions to the problems of battery condition monitoring, sunlight prediction, battery charge cycling, and optimum load management. These investigations were restricted to the case of photovoltaic (PV) charging of sealed lead-acid batteries.

Recurrent Backpropagation (RBP) networks were selected for noise detection, light prediction, and power cycling functions. Simulations were carried out with various size networks and minimization methods to establish the network solutions. Good results were achieved for battery condition monitoring using experimental data. For the sunlight prediction, a combination of detailed experimental local data and extensive summary sunlight data was used for model development. Using RBP networks, improvements over conventional statistical methods were realized.

A number of neural network optimization methods, including Hopfield, Boltzman, Mean Field Theory (MFT) and Free Energy Module (FEM) were evaluated for power optimization. MFT was selected for further investigations, and it was concluded that a neural network based optimizer can provide significant improvements in solar power system performance.

Introduction

Mariners depend on a variety of systems and devices for maritime navigation. The Canadian Coast Guard (CCG) places and maintains over 13,000 buoys. The largest of these are the 2.9 m diameter light and sound signal buoys placed in coastal waters. In 1990 there were 583 light and sound buoys in Canadian waters [1].

In the future, navigation buoys will be fitted with increasingly sophisticated sensing and communication systems. These will monitor a variety of state-of-health parameters such as water entry, light and flasher status, structural integrity, position, and depth; and communicate this information back to shore. The increase in on-board computer systems will enable the buoys to be used as "platforms-of-opportunity" for other data collection systems [2,3,4]. The additional systems will place increasing demands on the power system.

It should be possible to schedule power system demand to take advantage of present and projected future conditions. This would enable the extraction of more usable energy from a given power system, or conversely, the realization of smaller power systems and resulting cost reductions. Also, buoy authorities are adopting longer maintenance cycles to reduce overall maintenance cost. Hence, status monitoring and reporting becomes important.

This paper discusses a project to incorporate neural networks in an embedded controller to perform system health monitoring and optimization of buoy solar power systems. The objectives for the project were: investigate suitable network architectures and training methods; develop an experimental solution for the buoy model; and implement the solution in a prototype unit for field testing in a solar-powered buoy. The monitoring and control aspects addressed in the project are battery condition monitoring; prediction of sunshine based on previous weather; and optimization of battery charging and load control.

Neural Networks

Neural network computing is an alternative computational paradigm to the usual one (based on a programmed instruction sequence). It is inspired by knowledge from neuroscience, though it does not try to be biologically realistic in detail. Instead of having to specify every detail of a calculation, knowledge is embedded in a network by using a training set of representative examples [5].

Neural networks have been found to have advantages for certain types of problems embodied on our project. These include optimization, prediction, and sensor and data fusion. One of the characteristics of neural networks is their ability to generalize to new situations. This is felt to be one of the significant advantages for our application, where a controller may experience a variety of battery types, load scenarios and environmental conditions.

The neural network development system Xerion, from the Connectionist Research Group at the University of Toronto [6], was chosen for experimental development. The key criteria considered for evaluation of the neural development tools included the ability to support multiple paradigms; availability of source code generation; and analysis of network behaviour.

Experimental Development Model

An Experimental Development Model or XDM has been produced. The XDM, shown in Fig. 1, has been called the Power Monitoring and Optimization System, Buoy variant, or PMOS/B.

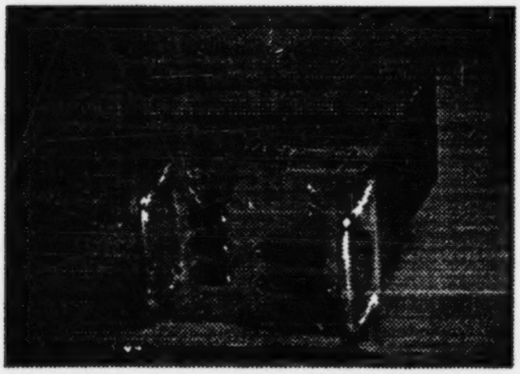


FIGURE 1: Power Monitoring and Optimization System, Buoy Variant.

Fig. 2 shows a block diagram of future navigation buoy systems. Instrumentation could include weather monitoring and oceanographic data collection systems. Typical sensors for such systems would be temperature, insolation (solar radiation) and other environmental sensors. Communications would be implemented using a radio or satellite link.

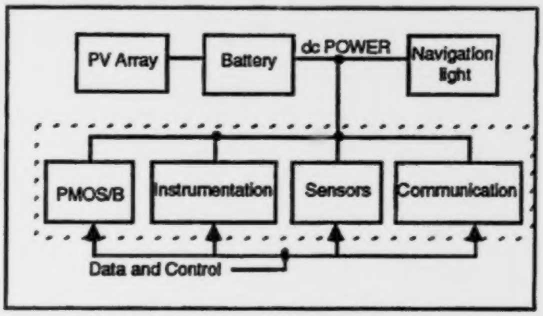


FIGURE 2: Navigation buoy system.

The PMOS/B will monitor the buoy power system and watch for the occurence of abnormal behavior and incipient or emerging faults. Corrective action may include shutting down non-critical activities, adjusting the schedule of events such as data collection or communications, and issuing a distress signal.

A functional block diagram of the PMOS/B is shown in Fig. 3. The PMOS/B uses the Harris RTX2000 microcontroller as its host processor. This is a high performance 16-bit microcontroller which uses a stack oriented, multiple bus architecture ideal for the implementation of the high level language Forth. The RTX2000 has several features such as one cycle subroutine calls and zero overhead returns which make the use of Forth on this microprocessor very desirable. The system controller has been implemented as a small high performance single board computer which serves as both a development platform and target processor. The entire RTX2000 ASIC bus is bussed out over a 50-pin connector as an interface to user-developed application circuitry.

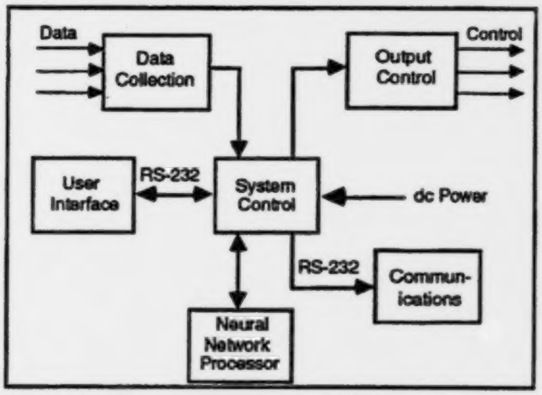


FIGURE 3: PMOS/B functional block diagram.

Preliminary Investigations

Some preliminary investigations were undertaken for simulator familiarization and selection of network types. The following optimization networks were first investigated:

- Boltzman Machine (BM)
- Mean Field Theory (MFT)
- Free Energy Module (FEM)

All three of these networks are derivatives of the original Hopfield [7] networks with added capabilities to avoid local minima and/or reduce computational requirements. The BM uses simulated annealing to eliminate local minima. Due to the large computational requirements of the BM, the MFT was developed which uses a deterministic algorithm to generate state transitions. The FEM uses a simplified training procedure for MFT networks which is fast, however, it is effective for binary classifications tasks only (i.e inputs must be 0 or 1).

It was clear from the initial tests that the BM networks were too computationally expensive to be of of practical use for our application. Tests with the MFT networks showed that they converged quickly and had minimal overall testing error. Tests with the FEM networks showed that they also perform well, however, they are limited to binary inputs. It was more likely that we would require analog inputs to our optimization network. After these initial tests we concentrated on using MFT networks for further optimization work.

A simple MFT power optimizing network was built. The network had three analog inputs (representing the probability of sunlight, battery condition, and battery charge) and three analog outputs (power to critical systems, power to lower priority systems, and power to perform communications).

The three inputs to the MFT network would be the outputs of Recurrent Back Propagation (RBP) [8] networks which would take time series of the sunlight, battery noise, and battery voltage. The sunlight input will indicate the probability of sunlight over the next period. The battery condition input is an indication of the potential for battery failure, based on time slices of battery noise. Battery charge, or energy stored in the battery, is represented by voltage. The time history of the voltage indicates the charging and discharging rates of battery which is also a measure of potential battery degradation.

A small amount of training data was generated which consisted of inputs and target outputs. The optimization network consisted of 3 input nodes, 8 hidden nodes, and 3 output nodes. With a sample set of 10 patterns it was trained for 500 iterations in approximately 1 minute. The resulting network had a very small error. When new testing data (inputs only) was applied it provided gradual changes in the outputs to optimize the power usage.

Three RBP networks consisting of 1 input node, 6 hidden nodes, and 1 output node were built. Training data was generated to represent typical daylight patterns, noise levels expected during battery failure, and voltage levels on battery. Two training runs were carried out with 10 time points and 100 time points. For these simple data sets the RBP networks performed well.

Neural Network Development

After a number of development iterations, the neural network power monitor evolved to the architecture shown in Fig. 4. The developed solution uses a combination of MFT and RBP networks. RBP networks were used to predict sunshine based on previous weather, monitor battery health, and monitor battery charge/discharge. An MFT network used the outputs of these networks, as well as other inputs, to perform the system power optimization. The following sections will discuss the development of the individual network blocks.

Light Prediction (RBP)

Neural Network Approach

Actual sunshine data was used to train and test light prediction networks. The first data (taken at Sable Island off the coast of Nova Scotia) contained average light levels for the hours of the day for each month of the year [9]. The data was averaged over the days of the month and over a large number of years. The data was normalized and formatted, then transferred to a Unix workstation. Networks for generating light predictions were constructed. These networks were RBP networks with one input, one output, and one hidden layer with a variable number of hidden nodes (4,8,16).

Various files were developed for each network including a layout file for graphic simulation, a network file, data input files, and overall control files. Numerous simulations were done with the original data serving as the training patterns and randomized data being used for testing purposes.

Most of the simulations were trained with a Quickprop training algorithm [6] to a maximum of 200 iterations. The basic result of these simulations was an overall 90% accuracy in predicting the light levels for the following hour. At this point, no statistical analysis was carried out to determine the worst case error, cumulative errors, variance, etc. There did not appear to be any significant increase in accuracy in going from 4 to 8 to 16 hidden nodes.

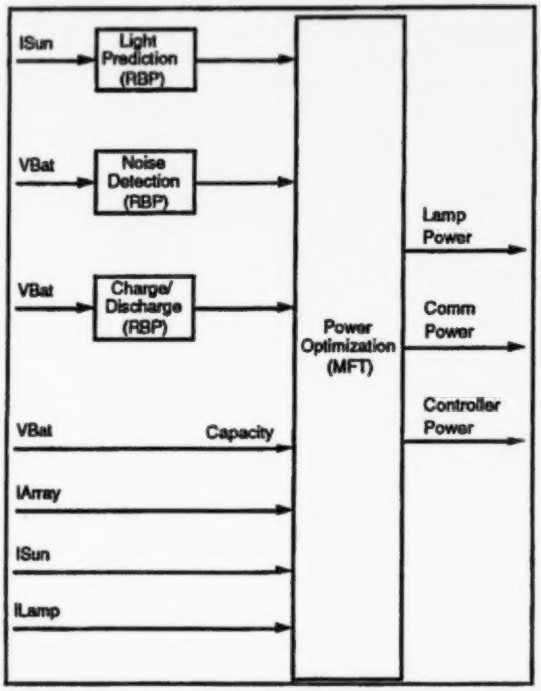


FIGURE 4: Neural network controller architecture

This original light data is averaged over many years, so most of the variances associated with local weather patterns, over short periods of time, have been eliminated. Therefore, this data is of limited use. A second database with daily light for every day from 1962-1991 provided a more suitable training set. This data, also for Sable Island, was obtained from Environment Canada on diskette.

Once again, numerous investigations were carried out with different numbers of hidden nodes (4, 8, 16, 32, 132). Most simulations were carried out with a maximum of 200 iterations. Various training algorithms were experimented with including Quickprop, Steepest descent, momentum descent, DeltaBarDelta (DBD) and Rudi's Conjugate Gradient (RCG) [6]. Initial training used 1988 data for training and 1989 data for testing. Subsequent training used both 1988 and 1989 data.

Some of the simulation results are as follows:

- average errors ranged from 6% - 14%,
- increasing the number of nodes reduced the errors,
- training with one pattern (year) resulted in larger testing errors in the second pattern (year),
- increasing the number of nodes to 132 resulted in large training times (1-2 minutes per iteration),
- increasing the number of iterations beyond 200 provided further reductions in errors, and
- the RCG training algorithm provided the best performance for most simulations.

Further simulations were carried out using a medium-size training database (10 years) with testing being carried out on one of the training years and validation being carried out on a year that was not part of the training set. Different size networks (4, 8, 16, 32 hidden nodes) were used to compare their performance. A summary of the results (averaged over 5 runs) can be seen in Table I.

TABLE I: Preliminary Light Prediction Results

Number of Hidden Nodes	4	8	16	32
Testing Error (20 trials)	15.3	13.2	12.3	12.4

As can be seen there are some reductions in overall error with larger hidden layers. The change from 16 to 32 hidden nodes resulted in a slight increase in overall error.

After settling on some network structures for the light simulations (1-16-1, 1-32-1), detailed batch simulations were carried out. These simulations were run with a large number of training trials (up to 2000) trying a number of different parameters (error minimization method, error radius, maximum weight change, added 2nd hidden layer). The training examples were increased to 20 years (1963-1983) to provide a larger training base. The best result achieved is a total testing error of 10.9 using a "Rudi Line Search" minimization method. This error translated into approximately 3% error per day on average.

Statistical Approach

As a comparison to the neural network solution, we also implemented a statistical predictor based on the approach developed by Zabara and Yianoulis [10]. This approach looks at day-to-day predictions and also at predictions of a sequence of k consecutive days of good or bad weather. The predictions are carried out for individual months thereby taking into account seasonal variations. The predictions for a sequence of days uses a stationary Markov chain approach which has also been used by a number of previous researchers.

The basis of our statistical approach to light prediction is the use of conditional probabilities of transitions from $0 \rightarrow 0$, $0 \rightarrow 1$, $1 \rightarrow 0$, and $1 \rightarrow 1$. The distinction between 0 and 1 is made by using a range of threshold values x between 0 and 1. Because you are comparing to some value x, we call these conditional probabilities rather than just straight probabilities. Programs were developed for calculating these quantities for each month of every year. The results of this analysis were compared to the analysis of Zabara. Our results were skewed to much lower light levels, however we did observe that there were in general higher conditional probabilities for 0→0 and 1→1 transitions than 0→1 and 1→0. This confirms that the amount of light in a given day is dependent on the light of the previous day (i.e. weather patterns). As mentioned the value of x is varied from 0 to 1. It was found that for high values of x the conditional probabilities were not statistically valid as the number of "1" values were so small that there were too few 1→1 and 1→0 transitions. For the summer months the x values could be much higher and still give statistically valid results (i.e there are much higher light levels in the summer months as compared to the winter months).

Once the conditional probabilities are calculated it is then possible to predict the sunlight for the next day. We used a Markov chain approximation which says that the probability of the next day being a one or zero is dependent on the previous k days as given by:

$$P_1(m,x) = P_{11}(m,x)**(k-1)[1-P_{11}(m,x)] \tag{1}$$

$$P_0(m,x) = [1-P_{01}(m,x)]**(k-1)P_{01}(m,x)] \tag{2}$$

where m is the month, and x is the threshold value for determining a 0 or 1.

What these equations say is that if we've had a number of 0's/1's there is decreasing probability (e.g $P_{11}**(k-1)$) that we will have a 0/1 tomorrow. However, given the higher conditional probabilities of 0→0 and 1→1 transitions, we start off with a good chance of having a sequence of 0's/1's.

A program was developed to read in present values of sunlight and predict tomorrow's sunlight based on the conditional probabilities and Markov chain probabilities. A typical sequence of predictions for the month of January can be seen in Table II. One can see that when we get a sequence of low values (Jan. 3-5) the prediction for the next day steadily grows. Similarly if we have a sequence of highs (Jan. 15-17) we will predict lower and lower values for the next day. The absolute value of the prediction is based on a probability of 0.5 equal to the average sunlight for that month (average over all years).

Predictions for all days of all years were generated using the statistical predictor discussed above and compared to predictions from our RBP neural networks. Programs were then written to determine the overall errors (sum of all errors) and absolute errors (sum of errors ignoring sign of error) for these two approaches. Additionally, errors were calculated based on the average monthly value of the light as the next days prediction. These errors can be seen in Table III.

TABLE II: Statistical Sunshine Predictions

	Present	Predicted
Jan 1	0.1533	0.0999
Jan 2	0.0267	0.0906
Jan 3	0.0000	0.1807
Jan 4	0.0000	0.2040
Jan 5	0.0000	0.2187
Jan 6	0.2067	0.0999
Jan 7	0.1000	0.0906
Jan 8	0.0000	0.1807
Jan 9	0.0067	0.2040
Jan 10	0.1533	0.0999
Jan 11	0.0067	0.0906
Jan 12	0.0000	0.1807
Jan 13	0.0067	0.2040
Jan 14	0.0000	0.2187
Jan 15	0.1200	0.2279
Jan 16	0.4333	0.0999
Jan 17	0.2067	0.0410
Jan 18	0.0000	0.0906

The important measure here is the absolute error as it determines on a day to day basis how far off we are in our predictions. We can see that the Neural Network (NN) approach was 9% better than the statistical approach and 6% better than the average value predictor.

This tells us that predicting sunshine is difficult and that our statistical approach wasn't as good as an average value approximation. There is room for improvement in the NN and statistical approaches, however, without optimizing either of the approaches the NN did perform significantly (9%) better than the statistical approach.

TABLE III: Sunshine Prediction Error

	Average	Statistical Predictor	Neural Network Predictor
Overall Error	0	66	171
Absolute Error	2562	2693	2417
Improvement	0%	-3%	6%

Noise detection (RBP)

Roberge has determined that the voltage noise produced in normal conditions of operation is a valid indicator of potential cell or battery problems [11,12]. A data acquisition experiment using National Instruments LabVIEW software and Macintosh IIx computer was set up to collect battery noise data. Voltage noise was measured using the methods of

Iverson and Roberge. In the measuring circuit a 1 MΩ resistor and 1 μF capacitor are used in series to block the dc voltage of the cell under study.

Two types of batteries were tested for faults using the noise measurement technique. The first type was a Panasonic LCR 12V6.5P 6.8 Ampere-hour (Ah) general purpose sealed lead acid battery. Measurements clearly indicated the presence of noise in a faulty cell, with almost no noise in a new cell. The noise patterns recorded for the Panasonic cell were used to train the networks to recognize the incipient failure of the cell. This data was in the form of two files, each with 2048 samples of the battery output voltage during discharge.

Simulations with RBP networks were carried out on the battery noise data. A number of simulations were carried out with different size networks. It was found that with a 1-8-1 network (1 input, 8 hidden, 1 output) we could completely distinguish between the two samples. After 50 training trials the total error was reduced from 735 to 9.9. This translates into less than 0.5% error on average for each of the 2048 samples.

The second battery type was a Yuasa NP24-12B 24 Ah sealed lead acid as used by the Canadian Coast Guard. Degradation in battery performance was observed by lower full charge voltage however no noise patterns were detected. Further studies are required to identify appropriate noise measurement techniques for these particular batteries.

Charge/Discharge (RBP)

For sealed lead-acid batteries, capacity diminishes as a function of the number of charge/discharge cycles and the depth of discharge. For example, if a battery is continually discharged to 20% of its capacity then it will have a lifetime of only 200 cycles. The derated battery capacity for a number of mappings used on the system simulator (described later) was calculated. From this it could be seen that the optimization strategy used affected battery capacity. To evaluate overall effectiveness of the optimization, derated battery capacity has to be considered.

We developed and trained RBP networks to derate battery capacity using the battery voltage time series as input to the network. The results were generally poor. The networks tended to average out the error instead of matching the expected values. It was concluded that a better approach to battery derating was to include a conventional power cycling procedure in the system simulator.

Power Optimization (MFT)

Using actual measurements (taken from a buoy solar power system in November) it was estimated that on a sunny day (relative total sunlight of 1.0) the solar cells could provide 16 Ah of stored energy. Using this data we did some analysis on the effectiveness of using sunlight data prediction in increasing the overall power usage of the buoys.

A battery storage capacity of 24 Ah was selected for system simulations, in order to exaggerate the battery charge fluctuations. The actual buoy power systems are currently very conservative, and much larger batteries are used. We first determined the continuous power dissipation level that could be supported with a storage capacity of 24 Ah, assuming we wanted a minimum reserve of 8 Ah. A C program was written to look at all the years (1963-1991) and find the worst case continuous level that could be supported. The minimum value, which occurred in January 1976, was found to be 1.0 Ah per day. A worst case design would use the minimum continuous power level at the expense of wasting sunlight energy that is provided.

The MFT network optimizes the power usage based primarily on inputs from the light predictor network and the present state-of-charge. A number of discrete power mappings were developed and used to train the MFT network. Table IV shows an example mapping, in which a target output power for training is given as a function of present charge and predicted sunlight. All numbers are expressed as coefficients of the actual maximum values. The numbers in parentheses are the actual values output by the trained network when presented with the respective inputs. The mappings train the networks with discrete values of present power and predicted light. When the input values are at some levels between these steps the network should produce output values that follow the general mapping function of the outputs.

TABLE IV: Sample Target and Actual Output Power Mapping

MAPPING1 Present Charge	0.00	0.25	0.50	0.75	1.00
Predicted Light					
0.00	0.1(0.06)	0.1(0.09)	0.2(0.17)	0.3(0.29)	0.4(0.56)
0.25	0.1(0.08)	0.1(0.13)	0.2(0.24)	0.4(0.46)	0.7(0.60)
0.50	0.1(0.11)	0.1(0.18)	0.3(0.34)	0.5(0.53)	0.8(0.71)
0.75	0.1(0.15)	0.1(0.26)	0.4(0.46)	0.7(0.67)	0.9(0.82)
1.00	0.1(0.20)	0.2(0.39)	0.4(0.58)	0.8(0.71)	1.0(0.86)

We tried a number of different mappings based on different assumptions about the overall power usage. For instance, in Table IV, for **mapping1** when the present power level gets low the output power is quickly reduced (i.e. when the present power is below 50% the output power is 10% of the maximum output power). This is based on the assumption that when the battery charge is high we may use a lot of power but once it starts to fall we must reduce our output to maintain a reserve against a long sequence of dark days.

We see that with the above mapping at a predicted light level of 1.0 and 0.0 present charge the output power is more than two times the target power. In this case the values are relatively small (10-20%) compared to the maximum output power. In later simulation of the entire system, we noticed that in some cases where there is low charge the networks will continue to output high power resulting in the battery being drained.

We used 13 mappings to try a variety of different power usage levels. Originally we tried automating the power level generation, however the combinatorial explosion of possibilities made this impractical. We decided to settle for a smaller group of potential mappings.

We built a number of MFT networks to perform the power optimization/mapping. We tried various networks with two inputs (present charge, predicted light) and one output (power to be used). We varied the number of hidden nodes and layers. Training and testing of the networks revealed that with 8 hidden nodes and one hidden layer we could get reasonable performance (similar accuracy as 16 and 32 hidden node networks). In general the mappings are fairly close for values ranging from 0.3 to 0.7, however, at the extremes the errors are generally larger.

System Integration and Simulation

We developed a C program called **calcpower** to integrate the light prediction network and power optimization network. This program takes the outputs of these two networks, as well as information on the battery charge and the observed light levels and computes new battery charge levels. This new charge level can then be fed back into the power optimization network to compute a new target power usage level. This whole feedback system runs from within Xerion. Xerion computes the energy usage, calls **calcpower** from a C-shell, reads in new charge and light prediction levels output from **calcpower**, and once again computes the energy usage. This allows us to essentially simulate the large feedback path encompassing the neural networks, battery, and solar panel input.

System simulations were run on the different MFT networks with various scaling factors. **Calcpower** has a Power Factor (PF) which allows the overall maximum power level to be adjusted. We have used different year datasets for light prediction and a number of different light prediction networks (i.e. different weight settings for the RBP network).

The following basic results have been found. For the 24 Ah battery used in our simulations it is possible to achieve 3 to 4 times the average output power. The cost is that the power dips will go below the continuous 1.0 Ah/day minimum mentioned previously. Of the simulations carried out the **mapping7** (see Table V) MFT network gave the best overall response, as shown in Table VI. This table indicates the number of days below a specified power output for various power factors, using both NN and statistical light predictors. The calculations were run on all years, a total of 10,585 days.

TABLE V: Mapping7 Values

MAPPING7 present charge / predicted light	0.00	0.25	0.50	0.75	1.00
0.00	0.10	0.10	0.10	0.50	1.00
0.25	0.10	0.10	0.10	0.60	1.00
0.50	0.10	0.10	0.10	0.70	1.00
0.75	0.10	0.10	0.10	0.80	1.00
1.00	0.10	0.10	0.10	0.80	1.00

TABLE VI: Full Network Response

	DAYS BELOW POWER OUTPUT			
	NN Light Predictor		Statistical Light Predictor	
Power Output	PF = 0.5	PF = 0.125	PF = 0.5	PF = 0.125
0.96 Ah/Day	1270	523	1282	499
0.72 Ah/Day	487	257	540	248
0.48 Ah/Day	62	45	68	39
0.24 Ah/Day	0	0	0	0
Average Power	4.0 Ah/Day	2.7 Ah/Day	4.0 Ah/Day	2.1 Ah/Day

Avg. Level for System Without Optimization = 1 Ah/Day

From the data it can be seen that for a power factor of 0.125, if we are willing to allow the power output to dip to 0.5 Ah/day for 45 days (0.4% of the time) we can provide average power 2.7 times that achievable with a continuous power consumption.

We can see that for a power factor of 0.125 the NN predictor provided 28% more power than the statistical predictor with less than a 5% increase in the number of days below 0.96 Ah/day. For a power factor of 0.50 the two predictors deliver the same average power with the statistical predictor having 3% more days below 0.96 Ah/day.

In general we see that the NN predictor is superior, particularly at the lower power factor. It also appears that for a given power factor the NN delivers more power, this is probably a result of what appears are large fluctuations in predictions as compared to those of the statistical predictor.

In an optimized power system, battery charging/discharging can have a significant effect on the performance of the system [13]. The authors found that this effect reduced the performance of the optimized system by approximately 20%.

Conclusions

TAVEL Limited, with the assistance of **Genlink Enterprises Ltd.**, has carried out a development project to investigate multi-paradigm neural networks in an embedded controller to perform system health monitoring and optimization of solar power systems. The solar-powered aids-to-navigation buoys maintained by the Canadian Coast Guard (CCG) provided a terrestrial model for the investigations.

The battery health network was able to distinguish a healthy battery from a defective battery with greater than 99.5% accuracy, based on time-series measurements of electrochemical noise. The light prediction network outperformed an average value predictor and a statistical approach by 6% and 9% respectively. It was found that power cycling was probably not a good problem for a neural network when compared to a conventional power cycling procedure. Using the power optimization we were able to obtain increases in available energy of 2 to 4 times over a continuous power consumption approach. Results using the neural network showed a 19% improvement over that using a statistical approach.

A neural network-based optimizer can provide significant improvements in solar power system performance. Prediction of the amount of sunlight to be expected is difficult. For our test case a neural network light predictor outperformed a statistical light predictor.

Acknowledgement

The project was funded through the Atlantic Canada Strategic Technologies in Automation and Robotics (STEAR) Initiative. The objective of the STEAR program is the development of the technological infrastructure required for the long-term support of the Space Station Freedom.

References

1. "List of Lights, Buoys and Fog Signals," Canadian Coast Guard, Transport Canada, Ottawa, 1991, TP39(4-7)E.
2. Capt. N. H.Keeler, "Maritime Future Navigation Needs and Plans," NAVIGATION: Journal of the Institute of Navigation, Vol. 34, No. 4, 1988.
3. Capt. Sir Miles Wingate, "The Future of Conventional Aids to Navigation," The Journal of Navigation, Vol. 39, 1986.
4. John C. Daidola, et al, "Buoy Technology Survey USCG Buoy Development Review," U.S. Coast Guard Research and Development Centre, Groton, Connecticut, October, 1990, CG-D-04-92.
5. J. Hertz, A. Krogh, R. G. Palmer, "Introduction to the Theory of Neural Computation, Lecture Notes Volume 1," Addison-Wesley Publishing Company, California, 1991.
6. D. van Camp, "Xerion Neural Network Simulator Users Guide," Department of Computer Science, University of Toronto, 1992.
7. E. W. Steeg, "Background: The Boltzmann Machine, MFT and FEM Algorithms," Department of Computer Science, University of Toronto, 1992.
8. *Darpa Neural Network Study*, Fairfax, VIR.:AFCEA International Press, 1988.
9. "Canadian Climate Normals, Volume 1, Solar Radiation 1951 - 1980," Environment Canada, 1982.
10. K. Zabara, P. Yianoulis, "Conditional Probabilities of Daily Relative Sunshine Data and the Dependence on the Weather of the Previous Day," *Solar Energy*, Vol. 48, No. 6, pp. 421-427, 1992.
11. P. Roberge, R. Beaudoin, G. Verville, J. Smit, "Voltage Noise Measurements on Sealed Lead-Acid Batteries," *Journal of Power Sources*, 27, pp. 177-186, 1989.
12. P. Roberge, E. Halliop, M. Farrington, "Monitoring Voltage Fluctutations for the Characterization of Lithium Cells," *Journal of Power Sources*, 34, pp. 233-241, 1991.
13. S. Harrington, J. Dunlop, "Battery Charge Controller Characteristics in Photovoltaic Systems," *IEEE Aerospace and Electronics Magazine*, pp. 15-21, August 1992.

902327

Powering Autonomous Underwater Vehicles with Aluminum

P.R. Roberge and J.P. Salvador
Department of Chemistry and Chemical Engineering
Royal Military College, Kingston, Ontario, Canada, K7K 5L0

R.G. Laidley
Directorate of Marine and Electrical Engineering
National Defence Headquarters, Ottawa, Ontario, Canada, K1A 0K2

#149659

Abstract

Government and industry research has developed a variety of autonomous undersea vehicles (AUVs) to perform subsea tasks free of bulky tether cables. The practical advantages of operating such vehicles over the more conventional tethered UVs are many but, above all, the cost benefits of operating missions with AUVs alone would make the consideration of this advanced technology an interesting proposition. AUVs are additionally independent of surface support and can transit faster then remotely operated vehicles (ROVs). Without any physical connection to the underwater vehicle, the surface vessel's navigation and operation is more relaxed and the submerged vehicle is not jerked or towed astray by the tether. Once operational experience builds trust in AUVs, it is predicted that many underwater tasks will be performed without the need for costly surface vessel support. This paper describes the typical mission profiles which can be required of an AUV and compare the cost of operating such vehicle with the cost of operating a more traditional ROV. The paper also demonstrates how the aluminum/oxygen power system is particularly well suited for its integration into AUVs. The advantages of using aluminum to power AUVs instead of using other powerful systems such as the Li or silver/zinc systems are described.

Introduction

Remotely operated vehicles (ROVs) and autonomous underwater vehicles (AUVs) are becoming more important to naval forces because of their enhanced ability to carry out complex underwater tasks with minimal human intervention. The miniaturization of computer electronics allows onboard computer controls to be preprogrammed to perform a variety of missions. A good example of the variety of military missions that could be performed by unmanned underwater vehicles (UUVs) has been described by the American Advanced Research Projects Agency (ARPA) in the course of a program originally established to demonstrate the practicality of UUVs in the Navy. The systems selected by the ARPA/Navy UUV program [1] were the tactical acoustic system (TAS), the mine search system (MSS), a mine avoidance aid, and the remote surveillance system (RSS), an underwater sensor array deployment vehicle. In 1992, a project was initiated at ARPA to demonstrate the use of a UUV for precise determination of underwater mine locations and bottom topography, designated the autonomous minehunting and mapping (AMM) system.

Operational Advantages of Using AUVs

Although ROVs can maneuver relatively freely at the end of an umbilical cord or theter that carries electrical power and communication links, there are definitive advantages to the use of the free swimming AUVs provided they carry on-board enough power to satisfy their missions [2]. AUVs are independent of surface support and can transit faster than ROVs, thus requiring reduced vessel time and crew [3]. The lack of a heavy tether to power an AUV eliminates expensive and high-maintenance cable handling equipment. The 21-inch diameter AUV prototype known as the XP-21 has been able to propel itself at a speed of 6 knots with only 2 horsepower of thrust.

Without any physical connection to the vehicle, the surface vessel's navigation watch circle is more relaxed and the submerged vehicle is not jerked or towed astray by the tether. An AUV can maneuver freely under acoustic control with a large horizontal offset from the operator. Below the surface, vehicle operations do not need to be interrupted due to bad weather because the AUV is unaffected by wind and sea conditions. Another advantage of AUVs over ROVs is their increased level of automation that requires only supervisory operator control. AUV operations require fewer crew members because they are less fatiguing for pilots and the overall job success is less dependent on operator skills. Once operational experience builds trust in AUVs, it is predicted that many tasks will be performed without the need for costly surface vessel support and personnel [3].

AUV Development

Over the past two decades a considerable amount of research has been invested toward a better understanding of the issues involved in AUV technology. A chronology of some of the more significant AUV development efforts is provided in Table I, which is based on information published in Busby's annual listing of underwater vehicles [4]. Most of the AUVs operating today appear to be limited in endurance to approximately 6 to 10 hours at relatively slow speeds. In early developmental programs, which focused primarily on software, endurance limitations were not a significant factor.

Secondary electrical storage batteries, despite endurance limitations, are a proven and reliable energy source which met the early needs of AUV developers. Presently, most AUVs operate on lead-acid or nickel-cadmium batteries, which severely limits their submerged operating time and range [2]. If the propulsion and electronic power requirements of AUVs could be satisfied with much higher density power sources, the AUV's future possibilities could be almost limitless. Such AUVs could be programmed to carry out multiple underwater tasks, report by acoustic telemetry or surface, report in by radio, receive new tasking and submerge to carry on.

Canadian AUV Specifications

The Department of National Defence (DND) has a future requirement for an AUV, and to meet this requirement, Defence Research Establishment Pacific (DREP) has undertaken research in different technology areas which are associated with either the mission or the vehicle. The mission of the vehicle is to deploy fiber optic cable under the ice between points up to 130 km apart, and then return autonomously to its starting point. The vehicle specifications are determined by this mission as well as by logistic and operational factors. A number of assumptions have been made which affect the mission performance. These are:

a) location of missions > 80 degrees north.
b) water depth will not exceed 600 meters.
c) surface is ice covered permanently.
d) ice keels will not normally protrude more than 60 meters from the surface.
e) ocean bottom is relatively flat.
f) tidal currents are typically < 0.25 knots.
g) ice at the terminus moves at < 10 km/day.
h) there is no ice movement at the launch point.

To allow for margins of error in navigation as well as the extra travel distance to avoid obstacle, the vehicle will be designed to be capable of a 400 km transit at speeds between 3 and 4 knots. At the outset of the program, in 1987, a feasibility study was conducted and it was decided that the power or energy source for the vehicle would not be an area of development because the requirements could be met with silver cathode batteries. But the recent progress in alternate power sources with higher endurance have encouraged a second investigation in the selection of power source for AUV to be carried out.

Energy Power Systems

The need for advanced energy systems to support the progress of both the UUV technology with only a communication umbilical cord and the AUV technology has been identified by many researchers and submersible vehicle developers. A study was recently undertaken in order to identify and evaluate energy systems potentially meeting the needs of future military, ocean science and commercial AUV applications [5]. The energy systems evaluated in this study are listed here under their main descriptor:

- **primary batteries**
  - lithium-thyonyl chloride (Li)
- **secondary batteries**
  - lead-acid (Pb)
  - silver-zinc (AZ)
  - nickel-cadmium (NC)
  - sodium-sulfur (Na)
  - nickel-metal hydride (NM)
  - silver-metal hydride (AM)
- **fuel cells and semi-fuel cells**
  - PEM hydrogen/oxygen fuel cell (FC)
  - aluminum-oxygen semi-fuel cell (Al)
- **closed cycle thermal systems**
  - closed cycle Diesel engine (CD)
  - closed cycle Sterling engine (CS)
  - stored chemical energy system (SC)

TABLE I AUV Development Chronology [4].

Year	Vehicle	Application	Mission	Depth (m)	Country
1963	SPURV I	Military	Under-ice mapping	3659	USA
1972	UARS	Military	Midwater research	457	USA
1973	SPURV II	Military	Midwater research	1524	USA
1975	SKAT	Academic	Ocean research	-	USSR
1979	EAVE EAST	Academic	Research testbed	150	USA
1979	EAVE WEST	Military	Research testbed	610	USA
1979	UFSS	Military	Search	357	USA
1980	SPAT	Industry	Acoustic training	240	USA
1980	PINGUIN IA	Industry	Search	200	Germany
1980	CSTV	Military	Submarine control	-	USA
1982	ROBOT II	Academic	Bottom surveys	91	USA
1982	ROVER	Academic	Structure inspection	100	UK
1982	B-1	Military	Flow characteristics	90	USA
1983	TM308	Industry	Structure inspection	400	Italy
1983	AUSS	Military	Search	6000	USA
1983	EPAULARD	Government	Bottom photography	6000	France
1983	AUV	Military	Hydrodynamic studies	-	USA
1984	ARCS	Military	Under-ice mapping	400	Canada
1986	ELIT	Government	Structure inspection	1000	France
1987	LSV	Military	Submarine testing	-	USA
1987	EAVE III	Academic	Research testbed	200	USA
1988	XP-21	Industry	Testbed	610	USA
1988	MUST	Industry	Testbed	610	USA
1989	SEA QUIRT	Academic	Survey	200	USA
1990	UUV	Military	Demonstration	450	USA

The data reported in this study were obtained from published sources and by direct correspondence with the system manufacturer. For a more consistent comparison, a baseline power level of 1 to 20 kW was adopted and, for systems requiring oxygen, a high pressure gas system was assumed.

Systems Effectiveness Evaluation

A figure-of-merit methodology was proposed and used to quantify the effectiveness of the selected energy systems for the AUV application. A series of relevant factors were identified, and subsequently broken down into sub-elements, to perform this evaluation. The main factors identified in this study were [5]:

- reliability
- reaction product impact
- operational experience
- specific energy
- energy density
- maintenance requirements
- logistic support requirements
- safety

To provide for some degree of objectivity in a process that can be prone to subjectivity, assessment criteria were defined for each of the precedent evaluation factors and each element was attributed a relative weight itself dependent on the requirements of specific AUV missions [5]. An overall weighted rating on a scale of 1 to 10 (with 10 most favorable) was derived for each system for each of the elements and of each factor. The results of the individual assessments were then combined into a weighted overall figure of merit effectiveness evaluation (Table II).

Candidate Energy System

As shown in Table II, the top three energy systems have nearly identical effectiveness ratings. While other energy systems investigated, such as the lead-acid battery, may offer advantages for some specific AUV applications, the three top energy systems appear to be the systems offering the highest effectiveness for future AUV missions.

TABLE II Energy Systems Effectiveness Evaluation Summary [5].

Factor	WT	Batteries							Fuel Cells		Thermal Systems		
		Li	Pb	AZ	NC	Na	NM	AM	Al	FC	CD	CS	SC
Operational													
Reliability	10	6	10	10	9	3	8	8	8	7	5	4	5
Reaction Product Impact	8	8	9	9	9	8	9	9	9	7	5	5	7
Recharge Impact	7	4	6	5	7	4	6	6	8	7	7	7	3
Experience	5	6	10	10	10	3	8	6	8	8	9	9	9
Weight & Volume													
Energy/Weight	8	8	1	3	1	3	2	4	6	7	3	3	10
Energy/Volume	8	10	1	3	1	2	3	8	5	5	3	3	10
Logistic													
Maintenance	5	8	9	9	10	4	9	9	7	4	4	4	4
Support	3	5	10	10	9	7	7	7	7	4	6	5	3
Safety	10	3	7	8	8	3	8	8	7	6	7	7	3
Weighted Total	64	411	425	460	434	248	420	466	462	403	340	327	391
Figure of Merit		6.42	6.64	7.19	6.78	3.88	6.56	7.28	7.22	6.30	5.31	5.11	6.11
Ranking		7	5	3	4	12	6	1	2	8	10	11	9

Silver-Zinc (Ag-Zn or AZ)

The Ag-Zn battery appears to be the battery of choice for many current AUV systems. This battery is available off the shelf and has a higher energy density and specific energy than many other batteries. High cost, limited cycle life and a long recharging process tend to reduce its effectiveness. Experience by many users of this battery indicates that the advertised cycle life (70-120 cycles) is often not achieved. For high discharge rate applications, such as in Navy torpedoes, the requirements call for 40 cycles over a 15 month period. In practice, observed cycle life has been as low as 8 cycles [5].

Silver-Metal Hydride (Ag-MH or AM)

The Ag-MH battery, a recent variant of the Ag-Zn battery, appears to offer significant improvements in energy density, specific energy and cycle life. Although current estimated costs are approximately twice that of the Ag-Zn battery, cycle life may be more than one order of magnitude greater than that experienced by the Ag-Zn system. But this battery is still under development and is not considered by manufacturers to be of the 'off-the-shelf' status. Hence, the system cannot be considered as a viable candidate for near term applications. When developed and available, this system has the potential to replace the Ag-Zn battery in many underwater vehicle applications [5].

Aluminum-Oxygen (Al-O_2 or Al)

Three types of Al-O_2 cells have been developed to date: a solids free system (SFS) which uses pumps and auxiliary equipment to manage reaction products, a solids self management system (SSMS), which eliminates much of the auxiliary equipment by allowing the reaction products to solidify in the cell enclosure during the discharge cycle, and a solids management system (SMS) which controls the precipitation of the alumina in a separate tank. The specific energy and energy density of all three systems appear very attractive for AUV applications.

Candidate Energy Summary

The Al-O_2 system appears to offer some advantages over Ag-Zn with respect to performance; however, the lack of operational experience in AUVs currently detracts from its evaluation. To help quantify the Al-O_2's potential as an AUV power source, the Canadian Navy is supporting an aluminum-oxygen demonstration to be undertaken in the spring of 1994. It will involve placing an aluminum-oxygen power source in Canada's ARCS vehicle and observing autonomous performance for thirty hours. The vehicle will have a solids free configuration and the oxygen will be stored on board as hydrogen peroxide.

Conclusion

UUV and AUV systems are likely to play an increasingly important role in future industrial, military and ocean science missions. The development of more capable and enduring AUV systems to meet the requirements of these missions is dependent, in part, on the development of improved energy systems. Past reliance on secondary batteries with relatively low energy densities and specific energies will not be adequate for the long range, extended duration AUV and UUV missions of the future.

Some major factors to be considered in the selection of energy systems for future AUV applications are: reliability, reaction product impact, recharge impact, operational experience, specific energy, energy density, maintenance, logistic support, safety and life-cycle costs. Operational reliability and safety are presently considered to be the most important of these factors. From an overall effectiveness consideration, the silver batteries and the aluminum-oxygen semi-fuel cell are evaluated as the top energy systems for AUV applications. A life cycle cost analysis of these systems is still needed to supplement such effectiveness evaluations.

The Canadian Navy near term tests of an AUV powered by an aluminum-oxygen semi-fuel cell will hopefully help to give this energy system the credibility required for it to become a major source of power in AUVs of the future.

References

1. G. Pappas, R. Rosenfeld and A. Beam, "The ARPA/NAVY Unmanned Udersea Vehicle Program," *Unmanned Systems*, Spring 1993, 41-44.
2. R. Laidley, "Canadian Designed Power Source for Unmanned Underwater Vehicles," 1993 Central Region Naval Engineering Seminar, Ottawa, Ontario, 1993, 1-3.
3. K. Collins, "Untethered AUV's Can Reduce Costs for Offshore Inspection Jobs," 25th Offshore Technology Conference, Houston, TX, 3-6 May 1993, 299-309.
4. F. Busby, "Undersea Vehicle Directory - 1990-91," Busby Associates, Arlington, VA, 1990.
5. G. Sedor, "A Comparative Analysis of Aluminum Based Energy Systems with Alternative Systems for Underwater Vehicle Applications," Marine System Eng. Lab., MSEL Report 93-01, Durham, NH, April 1993, 1-103.

POC 220

Simulation Based Artificial Intelligence for Intelligent Power System Protection

Mr. John Hensler [1]
HENSMAND Ltd., Ottawa

LCdr Mark Tinney [2]
Department of National Defence, NDHQ

#149660

Abstract

Electrical power and distribution systems on Navy ships are comprised of protection components designed to meet the needs of WW II technology. Remarkable advances in the technology of weapons, communications, and electrical power equipment have introduced new demands on circuit protection systems which are not being met by existing technology. The existing capability of protection circuits to respond to overloads, short circuits, reverse power, is no longer sufficient for modern warship systems. New warship protection schemes require a smart system which can discriminate between normal operation and true faults. In addition, protection systems can be designed to anticipate problems, react to remove the suspect component, and re-route power flow if necessary.

This paper will discuss the prospect of an intelligent protection system (IPS) based on computer simulations.

Clarification of Terms

-Artificial Intelligence is meant to describe a software product which is able to apply expert knowledge about the controlled system in a dynamic fashion without human intervention.

- Object oriented modelling describes the method of producing modular, generic and independent component simulators which are held together with other component simulators via I/O data files.

- Shallow knowledge base describes a rule based expert system technique.

- Deep knowledge base describes a dynamically adaptive computer simulation.

- Simulation describes the process of representing the actual system by its equations descriptive of the governing physics.

- Process control is intended to mean the management of outputs based on the behaviour of inputs.

In this paper simulation is used to enhance the Artificial Intelligence capability of the system, and as the next logical step up, AI is used to produce an improved Process Controller.

Introduction

Shipboard protection systems are made up of circuit breakers intended for industrial designs. In the past this was not a particular problem when a ship's distribution system mostly consisted of typical industrial quality components. Circuit faults were straight forward and predictable. If a circuit breaker responded incorrectly the problem was simple, there was something wrong with the pick up setting, or the breaker relaying circuit.

---

[1] HENSMAND Ltd., general manager
Phone: (613) 822-6922

[2] DND/DMEE 6, Electric Propulsion Systems
Phone: (819) 997-9377

Today the electrical distribution system onboard ships has changed considerably. The loads and interconnected power conditioning devices have become much more complex. Power electronic switching devices have introduced distortions which have adverse effects on most of the interconnected components in the distribution system including the protective devices. New weapon technologies, such as the electro-thermal gun will introduce significant pulse power demands on the distribution system. These power demands will require a high inertia generation system to sustain the voltage levels sufficiently that the magnet coils of the distribution system do not drop out on low voltage. At the same time we see the development of high speed low inertia generation systems for reduced weight and volume reasons, such as the Variable Speed Constant Frequency (VSCF) equipment. These VSCF systems will introduce new demands for the protection scheme, as they consist of high frequency generators connected to solid state converters. Typical circuit protection for such a converter consists of current limiting circuits designed to protect the semiconductor cells. These current limiting devices may also limit fault currents sufficiently to hamper the normal operation of standard protection devices. The same problem exists with small frequency changers which are now dispersed around the ship to meet the various 400 cycle requirements of the vessel. Their current limiting circuitry may inhibit the normal operation of circuit protection devices.

The protection systems found in modern warships have not kept pace with the rapid developments of the loads which they must supply. Such "modern" circuit protection systems have evolved only in that the monitoring circuitry has gone high tech, and hence offer better sensitivity while the control philosophy is basically unchanged from world war two schemes. There are two basic problems with the protection philosophy of existing warship protection systems. Typical protection systems are made up of independently acting circuit breakers without any reasonable form of centralized control and the operator of the vessel has no say in how the system should prioritize tripping events (i.e. the "decision" is made by hardware). In some cases operator interaction is highly desirable as it may be more acceptable to risk damaging the circuit and its components in favour of protecting ship's integrity. For example, in a ship which relies on thrusters for ship positioning the protection scheme will typically be set up to protect the thruster from overcurrent or destruction. But this philosophy ignores the scenario where the ship operator may be willing to risk the possibility of damaging the unit to avoid a much greater disaster such as a collision.

Conventional protection schemes monitor the typical fault parameters of overcurrent, undervoltage and reverse power and facilitate interfaces by their shunt trip and permissive relays. The control philosophy which governs their action is designed simply to protect the circuit and equipment to which they are connected. In an overcurrent situation, if the breaker closest to the fault does not react fast enough then the next breaker upstream will trip and as a result other components are taken off the line unnecessarily. This action may serve the protected circuit well but may lose sight of a greater purpose. In addition, the loss of large loads can lead to even greater problems with the supply network such as instability and trip of prime movers. This domino effect may occur at any time and can place the ship in danger.

The Intelligent Protection System (IPS) is intended to overcome these problems via the application of a centralized process controller philosophy. Before IPS can be described it is necessary to understand the electrical system and its requirements which have to be controlled.

System Characteristics

In any distribution system a fair amount of effort is required to set up the breakers so that they trip in correct sequence. Given that there are only a limited number of protection devices suitable for naval application, it is a challenge for the system designers to provide a fully coordinated protection within the constraints demanded by a shipboard system. The desired characteristics are:

- high speed clearance of faults while also releasing fault energy;
- maximum continuity of service under fault conditions;
- reliable operation of breakers;
- correct operation of breakers;

For cost reasons compromises are made and it is common to opt for less complicated systems with the resultant trade-off in slower response times. For example, the response times of typical circuit breakers are quite long, such as those shown below:

- instantaneous: 60-160 milliseconds
- overcurrent: 300-400 milliseconds
- overload: 15-30 seconds
- undervoltage: 500 milliseconds

In addition reverse power protection must be set quite slow to account for generator paralleling requirements and regeneration in ships with Electric Propulsion. These slow response times can lead to rapid degradation of the insulation values of the system in addition to damage to the contact components due to overheating and overvoltages. Even though the response times are slow the protection system components are still quire costly with high power feeder breakers costing hundreds of thousand of dollars each.

The issue of fault clearing speed is a function of three actions. The first is the time required to detect the fault and react, the second is the physical time it takes to open the contacts, and the third is the amount of time, it takes to extinguish the arc current. This paper is concerned mostly with the first aspect while consideration is also given to systems which address the other two.

Another issue which needs to be discussed is the fact that warship distribution systems are ungrounded. The reason for this design is to prevent the system from tripping out in the event of a single phase fault and prevent losing power to the rest of the network. This philosophy was fine for 1940s technology but does not serve the need of voltage sensitive loads such as computers or controllers which cannot tolerate a shift in the ground potential. On a modern warship these types of loads are in abundance. This has led to the need for a multitude of uninterruptable power supplies (UPS) to sustain the computer systems. Therefore selective tripping on ground faults is a desired feature, in these networks.

A Concept For The Future (Ref 1)

It is anticipated that the next generation of power plant protection systems will employ various levels of artificial intelligence (AI) techniques, and will adopt a process control philosophy for their implementation, and thus achieve intelligent control. The use of process control philosophy requires an integrated approach. In this sense the protection system shall take a coordinated response for both the electrical and the mechanical sides of the installation. This way the protection scheme will treat the power plant as a process with fuel as the input and electrical/mechanical power as the output.

To illustrate the point, consider a typical cycloconverter (CCV) electric propulsion control system. During power regeneration, in an existing protection scheme if the breaker connecting the CCV to the main bus should trip for any reason (such as a nuisance trip due to vibration) there would be no way to commutate off the thyristors and the control system would continue to gate the other cells. This would lead to a short circuit across the propulsion motor leading to a catastrophic failure of the shaft components under overtorque conditions. In the mean time the prime mover driving the generators would speed up on loss of load and may trip. The process would be further aggravated if someone would attempt to reestablish the supply and introduce the main bus to the three phase fault created by the CCV controller.

A process control approach for the same problem would incorporate a decision making feature based on the status of the entire plant. Upon sensing a loss of the sine waves one decision would be to suppress the firing of the thyristors. The second decision would be to adjust the fuel supply to the prime movers to prevent overspeed. Also the shaftline could be declutched if that capability were to exist, to offer further protection in the system.

The response times for faults can be dramatically improved through the use of an IPS system. This is because the issue of coordination no longer depends on tap settings and error margins. Instead, the IPS will be able to identify faults within the first rate of rise change and initiate response in microseconds. Reverse power operation will also be dramatically improved since the IPS would "know" if generators were being paralleled or if regeneration was taking place.

The IPS will allow the operators to select one of two modes of operation; a typical way to protect the circuit and the components under normal operation, and an emergency mode which will give priority to the protection of the vessel. In this mode the IPS will use a set of emergency parameters to allow equipment to be operated under higher levels of overload. This is necessary since in some cases it is more desirable to risk damaging the circuit and its components in favour of protecting the ship from damage.

IPS Architecture

The backbone of the IPS is an intelligent process controller which is continously updated with inputs from sensors throughout the system. The process controller derives its output signals from a simulator of the controlled system. In order to accommodate system changes due to battle damage or any other causes, the simulator is made to be a dynamically adaptive design by letting a shallow knowledge base initiate automatic I/O data file changes of the deep knowledge base routines, and hence allowing the expert knowledge of the system represented by these routines to generate conclusions with regards to the system's status.

IPS Development Considerations

The logic which will form the basis for the monitoring and control functions will be derived from computer simulations of the power plant.

To start, **mathematical models** will be made of all of the generators, loads, and interconnected equipment.

Then these models will be **translated into software modules**. Each module will form an independent simulator with input and output files.

The modules will be connected together in the required configuration and the system dynamics will be **simulated** under all normal and adverse conditions. This process will allow all possible fault characteristics to be identified. These fault characteristics will be captured and stored in look-up tables to be used later by the IPS.

Once the characteristics are identified, the next step is to **design a control system** which can recognize the normal and adverse characteristics of the distribution system and control the system as required.

The new control system design will be simulated and run against the power plant simulator to ensure that it performs as desired. The next step in the process is to actually build a working model of the control system. The performance of this hardware system can then be verified by running the power plant simulator against the working model. During this set to work procedure any problems in the control algorithms or the fault identification capabilities of the system can be identified and hence rectified early.

MODELLING: When building the model of the power plant it is important to include all the loads, load groups and represent them as accurately as possible. The reason is to identify how the loads will contribute to and be affected by any faults in the system. These loads include motors, transformers, UPS systems, frequency changers, as well as such equipment as changeover switches. It is also important to model the system at various load levels and have a good representation of how the generator sets respond to faults at various power demands. This is due to the fact that the levels of fault current will be determined by such parameters as the amount of inertia in the generation system, the amount of back-EMF in inductive loads, and the system power factor among others. In addition it is important to ascertain the lowest and highest subtransient fault currents including their DC shift for any condition and these are functions of the point on the waveform where the fault has initiated. Modelling the UPS's and frequency converters will aid in identifying how these systems will respond under fault conditions. For example for UPS systems such characteristics will be useful in determining which faults will cause the standby source to "kick-in". In the case of frequency converters, the model will also be able to identify how the faults will affect the output of these units, and ultimately how it will affect the loads supplied by the units.

SIMULATION: During the simulation the process control parameters will be based on long-time-constant (LTC) and short-time constant (STC) events. Normal system operation is typically described by LTC events, while system transients are typically found in STC events. There may be gray areas where the two groups overlap and these will require individual consideration. The IPS will monitor the system parameters and compare the systems actions and reactions to a "deep knowledge" base represented by the power plant simulator. Transient conditions will be stored in a "shallow knowledge" base in the form of look-up tables for fault signature comparison by matching characteristic signatures against a data base library. The overlapping parameters require a combination of the above two procedures. They will have to be established via a simulation and then moved into a rule-based expert module for real-time discrimination between faults and operational disturbances.

DESIGN CONSIDERATIONS: An important consideration for the IPS is to contain dynamic adaptive facilities so that on-board repairs, replacements, or alterations would not effect its operation. For example, consider a scenario where a drive motor is replaced by a machine of a different design, perhaps as an emergency or temporary measure. The process control will pulse the new motor and establish its characteristics by deriving the time-response of the new unit. This new data will be fed to a system simulation model and replace the control systems input data file for the motor. Then the simulator is run to derive the new set-points dead bands, ramping functions, to ensure stable operation. This information is compared against pre-set external acceptance limits. Upon finding an agreement the new control algorithms are implemented and the drive is ready for operation. Of course this will all happen at computer speed. Carrying this feature further, the system will also have to be able to dynamically adapt to changing system parameters as a result of aging, temperature. This adaptive nature of the IPS is seen to be the most important feature in designing survivability and damage control into the power plant, as it will reset the system when by-passes are activated.

The "learning" ability of the IPS will be utilized to trend the development of failures and be able to provide predictions when similar incidents develop in the future. Also the trends will be fed to the simulator and expert advice will be generated for implementing improvements for future avoidance.

In order to prevent nuisance actions of the IPS the process controller shall be able to work with noisy data or even make a good "guess" when there is a complete lack of data. This can be achieved in two stages. First, the data flow is passed through a Kalman filter to verify it against credibility criteria, and then a fault tolerant mode of operation is activated, if necessary. The Kalman filtering procedure uses a mixture of pre-set external thresholds for certain types of data. For other types a normalized value of previous information is relied upon as the baseline. In case of significant deviation, the input data is ignored and the previous sample is repeated, or interpolated based on a trend algorithm. If an error in data is persistent then the signal channel is declared to be faulty. In this event a default parameter is put in place of the signal value and a fail-safe mode is invoked.

To facilitate the use of a relatively simple CPU (central processing unit) such as the microprocessors found in standard industrial process controllers, it is proposed that frequency band multiplexing be relied upon for the monitoring of sensors. This is an important consideration when one realises that clearing of certain faults shall be affected within microseconds, and hence any time delays associated with data acquisition scanning rates simply could not be tolerated. As an example the IMCS equipment fitted on the Halifax class frigates operates with an update period of 500 ms, which would render the use of this device unacceptably slow for the purpose of developing an IPS (Ref. 2).

The operational philosophy of this IPS is depicted in Figure 1.

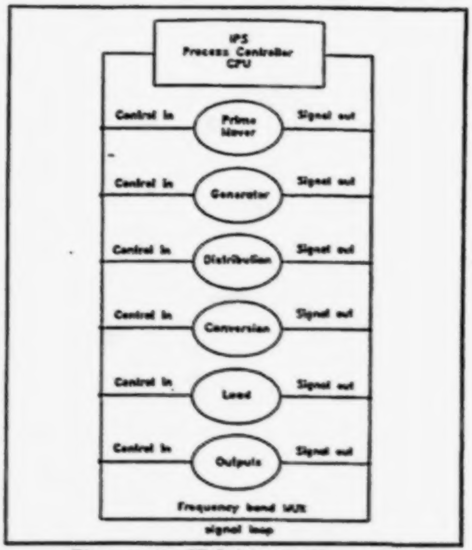


Figure 1: IPS block diagram

An important aspect of the IPS operation will depend on the overall design of the distribution system. The IPS will be capable of not only quickly clear faults but will also be able to re-route power. In order to do so it is necessary to design the distribution system with redundancy built in (Ref. 3). This does not necessarily mean that the system will become more complicated or require extra wire. A carefully designed distribution system can offer improved routing redundancy while actually leading to significant reductions in installed wiring, weight and cost.

The IPS design incorporates all of the sensing and control functions of the individual circuit breakers. Therefore the interruptors that would by used in an IPS system will be much simpler devices. All they require is a trip control system which is activated by the IPS. As such the IPS breakers will be less complicated than standard breakers but will still need to be of a rugged design to meet MIL specs.

Application of the IPS philosophy facilitates developing an extremely quick acting solid state circuit breaker (SCB) as depicted in Figure 2.

Normally the antiparallel SCR cells are kept open to allow the flow of sinusoidal power. Upon a fault sensed by the IPS the SCR cell gate pulses are removed within one half of a cycle and the rise of the fault current is reduced by the reactance as its flow is diverted to this device, and hence the system's fault energy is allowed to discharge.

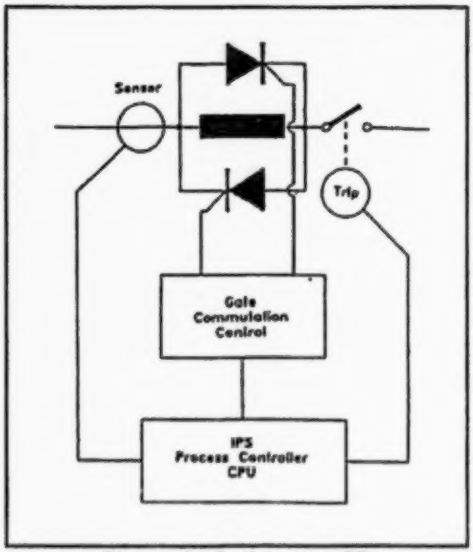


Figure 2: Solid state CB

When discussing costs of new systems one must look at the impact on operation and maintenance costs. Just the same way as automation in industry has lead to smaller workforces, the IPS system will allow a smaller complement of electricians. The goals of the IPS design are to take care of all of the distribution systems operations and be able to advise the operators of an impending problem as well as offer expert opinions with regards to system operational characteristics through the use of advisory messages. In this sense the IPS may also serve as a predictive maintenance tool and hence rather than reacting to problems when they occur, the operators would be able to correct problems before they become severe. This could undoubtedly reduce maintenance costs.

Conclusions

There are two conflicting demands placed on naval planners and designers. On the one hand warship distribution systems are becoming more complicated with transient inducing loads connected to the same network as sensitive loads. This demands a more sophisticated distribution system to ensure that a reliable supply is maintained under all conditions. Affordability is one of the most important considerations in todays economic environment, and hence it is imperative in electrical designs to use an integrated approach when designing a ship's distribution system to ensure that reliable operation is achieved in a cost effective fashion.

When one considers that the ultimate purpose of a warship is to go in harms way, survivability is a significant characteristic that must be carefully weighed against cost during the planning stages of a ship design. An integrated Intelligent Protection System along with a carefully designed distribution system can be considered as a way to achieve a higher level of reliability without incurring large cost increases. This centralized process control system has the potential to limit the effects and propagation of system damage through faster operation and can maintain a higher level of system availability by ensuring that only the necessary breakers are tripped under fault conditions, as well as by re-routing power wherever possible. It could also add a greater amount of stability to the system by controlling not only the protective devices but also the generator prime movers and some times the output as well (such as for electric propulsion systems).

Computer simulations are extensively used by DND. Such tools can not only serve during ship design, but can also form the backbone of a cost effective and affordable artificial intelligent control module for power plant protection systems. The modular, object oriented simulation routines used for electric propulsion system studies by DND can serve as process controllers capable to run an integrated and coordinated IPS.

The system operators will benefit from an intelligent system which will advise them of impending problems. Adoption of the IPS philosophy can lead to reduced operating costs in the form of a smaller ship's complement, and reduced maintenance costs, as a result of changing over to a predictive approach from the presently used preventive maintenance philosophy.

References

1. IMCA corporate material
   "Intelligent Machinery Condition Analyzer"
   by: HENSMAND Ltd. - Integral Dynamics Division
2. DND document
   "CPF Failure Mode Analysis"
   by: HENSMAND Ltd. - Integral Dynamics Division
3. "Zonal Electrical Distribution Systems"
   by: C.R. Petry and J.W. Rumburg, Naval Engineering Journal, May 1993
4. Polar 8 design workbook
   "High voltage propulsion systems integration"
   by: HENSMAND Ltd. - Integral Dynamics Division

An Inductive Learning Approach to SARSAT Confidence Factors

Ivan Taylor
Operational Research Branch
Air Transport Group Headquarters
Astra, Ontario, K0K 1B0

Hong Zhao
ABIT System Incorporated
661 Huron Street, Apartment 105
Toronto, Ontario, M5R 2R8

#149661

Abstract

An inductive learning approach is developed for the Search and Rescue Satellite Aided Tracking (SARSAT) Confidence Factor Method. Qualitative distinctions are derived for the SARSAT signal parameters that demonstrated significant discriminating capability. The SARSAT data was transformed according to those qualitative distinctions and then analyzed to obtain a decision tree and rule base with which to classify the observations. The accuracy of this method was comparable to other techniques that have been examined in previous studies. The appealing feature of this technique is that the decision rules are intuitively comprehensible and simple to apply.

Introduction

The Search and Rescue Satellite Aided Tracking (SARSAT) System uses polar-orbiting satellites to monitor the surface of the earth for Emergency Locator Transmitter (ELT) signals. If one is detected, its signal is retransmitted to ground facilities, called Local User Terminals (LUTs) for processing. The location of the ELT is determined by the LUT using a Doppler curve-fitting procedure and then this location is sent to the Mission Control Centre (MCC) for retransmission to a foreign MCC or a National Rescue Coordination Centre (RCC) for action.

The SARSAT System has a particular problem with false alerts on the 121.5 MHz frequency. This frequency is reserved for emergency signals, however, quite often inadvertent voice communication on the frequency will be picked up by the satellite and processed as an ELT signal. The Operational Research Branch at Air Transport Group developed a heuristic technique to evaluate the signal quality that is used to screen out these false alerts. This technique is called the Confidence Factor Method [1, 2 and 3] and it has been implemented in the LUTs.

Unfortunately, the current heuristic is not well understood and is based on empirical data rather than a sound theoretical foundation. Furthermore, the results are far from ideal [4]. The Operational Research Branch has been investigating other methods that might replace the current heuristic. Any new method should have better performance, be easier to understand and have a more rigorous theoretical foundation. An earlier study [5] examined an approach from artificial intelligence literature called neural networks. Although the results were promising this technique is not easy to understand and would be basically a "black box" approach. The techniques from cluster analysis and statistical discriminant analysis were also examined [6]. These techniques produced results comparable to the neural network approach and had a number of favourable features that deserve further study.

In the following study, we will examine the technique from artificial intelligence called inductive learning. We will interpret the data parameters using techniques from "fuzzy set" theory [7] to determine intervals in which the parameters could be considered to have qualitative distinction. We will apply an inductive learning algorithm to generate intuitively understandable rules that can be used to classify the SARSAT data into confidence factor categories. We will evaluate the results of the inductive learning approach compared to the other methods that have been already examined and we will draw some conclusions about the appropriateness of further research into this technique.

Data Analysis

In [6], the parameters available from the SARSAT system were evaluated for their discriminating capability. There were nine parameters examined: the satellite, the major axis of the error ellipse, the minor axis of the error ellipse, the probability that attempts to discriminate the real solution from its image, the signal bias, the satellite cross track angle, the Doppler curve-fit standard deviation, the number of points in the signal and the normalized quality factor of the signal.

The data set consisted of 345 traced ELT cases which was divided into 173 cases for the training set and 172 cases for the test set. The data in the training set was further divided into three confidence factor categories corresponding to the error in the location estimate. Category 'A' included 83 cases that had a location error less than five nautical miles. Category 'B' included 61 cases with a location error between five and twenty nautical miles. Category 'CD' contained 29 cases with location error greater than 20 nautical miles.

An earlier study [6] provided the results of the data analysis in graphical form. Of the nine parameters, the major and minor axes, the standard deviation and the number of points were considered to have sufficient discriminating power to be included in the inductive learning experiment. That is, the percentile graphs demonstrated a significant separation in the curves for each category. It was necessary to develop intervals to describe these parameters in qualitative terms for the inductive learning algorithm. First the values for the major, the minor and the standard deviation were transformed using the natural logarithm function to allow further statistical modelling using the Normal probability distribution. Probability density functions were developed using the means and standard deviations in Table I. The values at which the density functions overlap were taken as the delimiters for the qualitative regions: good, fair and poor. Table II provides the delimiters that were used to categorize the data for further analysis using the inductive learning technique.

TABLE I: Means and Standard Deviations for SARSAT Parameters

PARAMETER	CATEGORY	MEAN	STANDARD DEVIATION
LN(MAJOR)	A	1.5535	0.7868
	B	2.2450	0.7811
	CD	3.1378	0.9377
LN(MINOR)	A	0.7300	0.6649
	B	1.3151	0.7148
	CD	2.0431	0.7045
LN(STANDARD DEVIATION)	A	1.9328	0.4820
	B	2.2189	0.4704
	CD	2.6516	0.6189
POINTS	A	255	85
	B	213	85
	CD	163	79

TABLE II: Intervals for Categorizing the SARSAT Parameters

PARAMETER	GOOD	FAIR	POOR
MAJOR	$X \leq 6.6$	$6.6 < X < 16.0$	$X \geq 16.0$
MINOR	$X \leq 2.9$	$2.9 < X < 5.3$	$X \geq 5.3$
STANDARD DEVIATION	$X \leq 7.8$	$7.8 < X < 13.1$	$X \geq 13.1$
POINTS	$X \geq 234$	$197 < X < 234$	$X \leq 197$

For example, SARSAT signal with a major axis of 7.0, a minor axis of 2.0, a standard deviation of 10.0 and 175 points in the signal would be considered to have a 'fair' major parameter, a 'good' minor parameter, a 'fair' standard deviation and a 'poor' number of points.

As we shall see in the next section, the training data sets was not able to support this level of distinction and in fact, we used only two regions: 'poor' and 'not poor'. That is, we grouped the 'good' and the 'fair' regions together to be called 'not poor'.

Inductive Learning

Inductive learning algorithms, such as Quinlan's ID3 [8], are based on the analysis of discrete data. In the SARSAT data files, we have continuous variables. In the previous section, we converted the continuous variables into discrete symbolic variables using an approximation to the fuzzy membership function.

Inductive learning algorithms produce IF-THEN rules based on an analysis of patterns in the training set. For example, based on our symbolic variables a rule might be IF MAJOR IS 'NOT POOR' AND MINOR IS 'NOT POOR' THEN CATEGORY IS 'A'.

These rules make up a decision tree in which an expert system could be developed to trace through each case to determine the appropriate response. However, we wish to ensure that in every case, the expert system will produce a response. The finer the distinction in the variables, the more branches there are in the decision tree and the more examples one needs to develop the decision rules. We found when we applied the inductive learning algorithm that the distinctions of 'good', 'fair' and 'poor' produced many situations where the training set could not support the development of a specific rule. Therefore, we grouped the data into more general classes. Namely, the 'good' and the 'fair' intervals were combined and described as 'not poor'. This allowed a full decision tree while maintaining the symbolic meaning of the parameter values.

A second issue that arose in the development of the inductive learning rules involved the situation in which there was conflict in the training set. For example, there are many situations in which two cases with the same parameter values have different resulting categories in the training set. To resolve this problem, we took the approach of a "voting" system in which the category which was most often observed was taken as the prediction in this situation.

Finally, we examined alternative inductive learning schemes. The latest ID3 algorithm developed by Quinlan [9] produced unsatisfactory results when applied to the test set (i.e. only 46% accuracy). A different version of Quinlan's algorithm developed by Paterson, Niblett and Shapiro, called the Analog Concept Learning System (ACLS) [10], produced better results of about 55% accuracy and was chosen as the inductive learning scheme for this study.

TABLE III: Rule Base from Inductive Learning

RULE	MAJOR	MINOR	STANDARD DEVIATION	NUMBER OF POINTS	PREDICTION
1	NOT POOR	NOT POOR	-	-	A
2	NOT POOR	POOR	NOT POOR	NOT POOR	A
3	NOT POOR	POOR	NOT POOR	POOR	B
4	NOT POOR	POOR	POOR	-	B
5	POOR	NOT POOR	-	-	B
6	POOR	POOR	-	NOT POOR	B
7	POOR	POOR	-	POOR	CD

TABLE IV: Accuracy of Inductive Learning Approach

PREDICTED CATEGORY	ACTUAL CATEGORY			TOTAL
	A	B	CD	
A	59	37	8	104
B	4	15	9	28
CD	4	16	20	40
TOTALS	67	68	37	172

In [11] and [12], we provide a discussion of the specific procedures that were used to develop the rule base using dBase IV programs to analyze the data set and a program called SuperExpert to implement the ACLS process. The rule base resulting from the ACLS algorithm consists of seven rules as shown in Table III.

Thus if the parameters of the SARSAT case are such that MAJOR is 'poor', MINOR is 'poor', STANDARD DEVIATION is 'not poor' and POINTS is 'not poor', then Rule 6 would apply and the prediction would be Confidence Factor 'B'. Notice that in Rule 6 the STANDARD DEVIATION could have been 'poor' or 'not poor' and the Confidence Factor prediction would have been the same (i.e 'B').

Results

Table IV provides the results of the application of these rules to the test set. We can see that this method does a good job of identifying the A category cases and a reasonable job of identifying the CD category cases. It appears to have a great deal of difficulty determining the category B cases. This is possibly a result of a lack of distinction in the data set for category B cases. The overall accuracy is 55% which is comparable to other techniques that have been examined [6].

When we compare these results to those in earlier studies, we see that this technique appears to have some problems with over-confidence. That is, it appears to often classify mediocre observations as very good solutions. More balanced results have been obtained by the discriminant analysis technique that was developed in [13].

Concluding Material

An inductive learning approach was applied to the SARSAT confidence factor data set and a small set of intuitively understandable rules was derived to classify the observations into one of the confidence factor categories. The procedure that would be used to apply inductive learning involves:

a. the development of symbolic representation of the data using an approximation to a fuzzy membership function;
b. the transformation of the training data set into these symbolic classes;
c. the application of the ACLS inductive learning scheme to develop a rule base;
d. the resolution of problems such as 'null-nodes' on the decision tree and conflict in the training set; and
e. the application of the rule base to the test set and analysis of the results.

A number of issues arose in the application of inductive learning to the SARSAT confidence factor method. Although these were resolved in our research effort, the solutions were somewhat expedient and unsatisfactory. The capability to utilize the full discriminating power of the parameter classes might be resolved with a new and larger data set.

The value of intuitive rules in the determination of the confidence factor probably allows for a better understanding of the SARSAT Classification process. Although, the prediction accuracy is comparable to other methods, we would not recommend that this method be implemented at this time because the results are not balanced. We suggest that further investigation might attempt to place a possibility distribution on the confidence factor prediction using techniques from "Fuzzy" Cluster Analysis [14].

References

1. Charles J. Hunter, "A Critical Review of COSPAS/SARSAT Data Categorization and location Accuracy Models", ATGOR[1] Staff Note 1/87, August 1987.

2. Christopher M. Riehl, "COSPAS/SARSAT Canadian Location Accuracy Models for 121.5 MHz Distress Signals", ATGOR Staff Note 3/87, August 1987.

3. Charles J. Hunter, "COSPAS/SARSAT Canadian Location Accuracy Models for 121.5 MHz Distress Beacons: Revisited", ATGOR Staff Note 1/88, April 1988.

4. Ivan Taylor and Michel Vigneault, "Preliminary Research on a New Confidence Factor Methodology", ATGOR Staff Note 4/91, August 1991.

5. Ivan Taylor and Michel Vigneault, "A Neural Network Application to Search and Rescue Satellite Aided Tracking (SARSAT)", ATGOR Working Paper 6/92, March 1992.

6. Ivan Taylor and Michel Vigneault, "Evaluating New Confidence Factor Methods", ATGOR Working Paper 13/92, September 1992.

7. Constantin V. Negoita, "Expert System and Fuzzy Systems", Benjamin/Cummings Publishing Company, Menlo Park, California, 1985.

8. J.R. Quinlan, "Learning Efficient Classification Procedures and Their Application to Chess End Games, in Machine Learning: An Artificial Intelligence Approach", edited by R.S. Michalski, J.G. Carbonell and T.M. Mitchell, Tioga Publishing Company, Palo Alto, 1983.

9. J. Ross Quilan, "C4.5 Programs for Machine Learning", Morgan Kaufmann Publishers, San Mateo, California, 1992.

10. A. Paterson, T. Niblett and A. Shapiro, "ACLS User Manual", Intelligent Terminals Limited., Glasgow 1982.

11. Ivan Taylor and Hong Zhao, "An Inductive Learning approach to SARSAT Confidence Factors", ATGOR Working Paper 6/93, April 1993.

12. Hong Zhao and Michel Vigneault, "Programs for Inductive Learning with Fuzzy Semantic Intervals, Addendum to ATGOR Working Paper 6/93?April 1993.

13. I.W. Taylor, Attaching Probability to the SARSAT Confidence Factor, ATGOR Research Note 3/93, March 1993.

14. Meeting Operational Research Advisor/Dr. I.B. Turksen, Abit Systems Incorporated and University of Toronto, Department of Industrial Engineering, 22 March 1993.

---

[1] ATGOR = Air Transport Group Operational Research

90C218

DND Knowledge-Based Systems and Robotics Workshops, November 1993

A Plan Evaluation Methodology for Military Command and Control

Major Robert Weisman[1]
Directorate of Information Systems Plans
National Defence Headquarters,
5th Floor Constitution Building 305 Rideau
Ottawa, Ontario

Dr. Jean Fugere[2]
Department of Mathematics and Computer Science
Royal Military College of Canada, Kingston, Ontario, K7K5L0

#149662

Abstract

One of the principal Military Command and Control processes is the formulation of a plan in what is called the Estimate Process. Key sub-processes include the evaluation of a possible course of action and the selection of the best course of action to act as the nucleus of a plan. This paper proposes a methodology which will enable automated assistance for the evaluation of a Course of Action.

The Plan Evaluation Methodology is based upon a detailed Conceptual Model, which provides a structure for the knowledge; a specialized Constraint Language, which facilitates the precise expression of the knowledge; and a Knowledge Acquisition Technique, which provides a systematic means of acquiring the knowledge. Part of the latter technique includes the establishment of an Action Model, based on a standard military operation such as "Brigade Counterattack", and a Factor Matrix where the elements of the Action Model are assessed with respect to the factors commonly considered by military commanders (eg."Enemy Air Capability") in the estimate process.

For each element in the Factor Matrix, the rules used by experts to determine the relative merit of the various parts of the Action Model are acquired using a "Possible Advantage-Disadvantage" metric, which was developed to approximate the military way of assessing a course of action. The term "Possible" indicates that these general rules are often encapsulated in doctrine and are not always applicable.

Afterwards each factor is examined and the expert is asked to determine its applicability using a "Factor Relevance" Metric. The Factor Relevance is then combined with the derived "Possible Advantage/Disadvantage" value to come up with a clear measurement of "Advantage/Disadvantage".

All of the derived Advantages/ Disadvantages are then amalgamated using both a "Least Risk" Metric, which classifies the course of action with respect to its least Advantageous State, and a "Best Advantage" Metric, which assesses the course of action using a weighted sum of all the Advantageous/ Disadvantageous states. These assessments form the basis for comparing various Courses of Action and selecting the best one.

This paper will describe the acquisition of the relevant knowledge and the results of prototyping of this plan evaluation technique on Personal Computers using VPExpert and on workstations using the Automated Reasoning Tool software.

---

[1] (613)996-9836

[2] (613)541-6475

Introduction

Few of the fielded decision support systems actually assist in command and control related activities rather they focus on the organization and presentation of key information. To build a system which has some, however basic, understanding of what is going on, requires the definition of a detailed conceptual model of reality which will permit the acquisition, storage and manipulation of the requisite knowledge. The elaboration of such a model requires a sound understanding of both the processes involved in command and control and the development of complex software programs as well as a solid insight into how experts handle emergency situations. After discussing the fundamental Command and Control and defining some of the basic elements of a conceptual model, this paper will propose an effective and simple means of evaluating a plan. The paper will end with a discussion of some of the prototypes and some of the areas where further research is required.

The Military Command and Control Process

Military studies in Command and Control have identified three primary activities namely Planning, Directing and Monitoring. (TCCCS 1989). The primary activity which concerns us is the Planning process which involves carrying out an Estimate of the Situation and the determination of a Course of Action. An Estimate includes an evaluation of the current situation taking into account all of the aspects which might influence the most effective execution of an assigned, or assumed, task. Possible ways of executing the assigned, or assumed, task(s) are called courses of actions. Therefore the Planning Process has the commander carrying out the Estimate process to determine what tasks have to be carried out, to conceive of several courses of action which could complete these tasks and then to compare these courses of action in order to select the one which offers the best chance of success. This Best Course of Action is then expanded into an outline plan.

The Directing process takes this outline plan and adds on the requisite detail for the completion and dissemination of a formalized operations order which can be unambiguously understood by those personnel charged with the execution of the plan. The Monitoring process involves the collection, collation and assessment of all plan related information and ensuring that reality conforms both to the details elaborated in the operations order (eg. Unit A is at Point B at Time C) as well as those fundamental assumptions upon which the plan is based (eg. There will be no precipitation tonight).

Any system designed to assist a commander should be able to eventually address all three of these processes.

Selected Conceptual Model

An object-based conceptual model was selected, discussed at length in Weisman 1992, and modified to suit the command and control problem domain. Rather than create a unique conceptual model (as was initially proposed in Weisman et Al 1990a, 1990b), the model adopted, as a basis, the conventions of the widely accepted Object-Oriented Analysis (OOA) (Coad and Yourdon 1990,1991) methodology. This decision facilitates the use of the model by knowledge engineers not fully conversant with Command and Control and also there exist Computer-Assisted Software Engineering (CASE) Tools which make the analysis, design, implementation (Coad and Yourdon 1993) and documentation much easier. In short flexible systems could be conceived and fielded much faster than previously possible. Another advantage of adopting this conceptual model at this level of abstraction is that the actual design and implementation issues can be independent of the analysis/ knowledge acquisition. For example the analysis used for some of the prototypes was actually implemented using an Expert Systems Shell in conjunction with spreadsheets and associated programs written in C and FORTRAN; there is no need to design and code in an object-oriented language.

For the purposes of this paper only certain key objects will be described, with more detail being available in Weisman 1992. There are two fundamental categories of objects, basically those used to represent objects existing in the real world and those used to store the knowledge required for the formulation of a plan. These two categories of objects will now be briefly explained using Object Oriented Analysis (OOA) Terms from Coad and Yourdon (1990, 1991). Of note is the term "Service" which is defined as a specific behaviour that an Object is responsible for exhibiting (Coad and Yourdon 1991). It is analogous to the term "Method" used in Object-Oriented Programming.

Real World Objects

In order to reason about plans, there must first be defined a set of basic objects whose definition will permit the eventual elaboration of a plan. In our case these subjects are agents, actions, planning objects, and physical objects. The associated terms and symbology are described in Figure 1. Note that the symbology has been modified to make the identification of objects belonging to these subjects easier. This feature facilitates the initial representation and validation of the knowledge during the acquisition phase.

Physical Objects are those which can be directly mapped into an object in the Physical World. In our context, an example of a Physical Object could be a Tank or a Terrain Feature. Physical Objects are specialized as either Human, Materiel, Terrain or Climate Objects. Materiel Objects are used in the execution of actions. They are further specialized as either Equipment Objects which are those non-human objects directly used in the execution of actions, such as a Tank, or as Supply Objects, such as Fuel, which enable the functioning of both human and equipment objects. Terrain Objects refer to those geographic features of significance, such as a marsh. A Climate object will encapsulate the various climatic data that will be required by the services belonging to the other objects.

Planning Objects are abstractions used by planners to spatially constrain the specification, and subsequent execution, of an Action. An example is the designation of the Planning Object Class "Objective". An instance of objective could be "Objective Bravo" which is indicated, by the commander, as a circle inscribed on a map. Note that it is the topographical features within and outside the objective area that give that abstract circle any meaning in the physical world. The Action "Capture Objective Bravo" would be assigned as a task (or goal) to the agent "2nd Battalion Royal Canadian Regiment".

Agents are those objects which plan and execute actions. They are a grouping of physical objects, in particular personnel, equipment and supply class-&-objects, combined for the purpose of carrying out a set of assigned actions.

An Action Object describes an action which has to be either planned and executed or assigned as a task by an agent. An Action Object has instance connections to both Agent Objects (the Agent assigning the Action and the Agent executing the Action) and to Planning Objects which are the object of the Action. An Action is composed of a non-unique series of sub-actions, with the recursion terminating at a level of abstraction deemed suitable by an expert. An ideal decomposition, based upon an expert's opinion, for a specific action is represented in an Action Model.

Other Objects Used for Planning Purposes

The following Conceptual Model Objects directly cope with the aspect of deriving and evaluating a plan. The goal of these objects is to store and structure the requisite information and expert knowledge. There are two basic types of object, namely the Models and the so-called Containers. The models contain the expert's/doctrinal default knowledge to complete a pre-defined action/plan/estimate. The containers include and coordinate instances of the various models and incorporate the user's and/or contextual modifications of these models.

The basic Model Objects are the Action, Estimate and Plan Models. Let us assume that for every basic military operation, such as "Defence", there exists an Action Model, albeit at varying levels of abstraction (eg. Division, Brigade, Battalion), an Estimate Model which contains the knowledge required to both assist and evaluate the possible Courses of Action, and a Plan Model which contains the knowledge required to transform the best course of action into a complete plan.

The next category of objects are partially predicated on the assumption that an Agent is normally assigned more than one Action, each of which has an Action, Plan and Estimate Model. These basic Container Objects are a Plan, an Estimate, and a Course of Action. An instance of a Plan includes and merges the aspects of one or more Plan Models. An Estimate Object contains and merges the aspects of one or more Estimate Models. A Course of Action represents the mini-

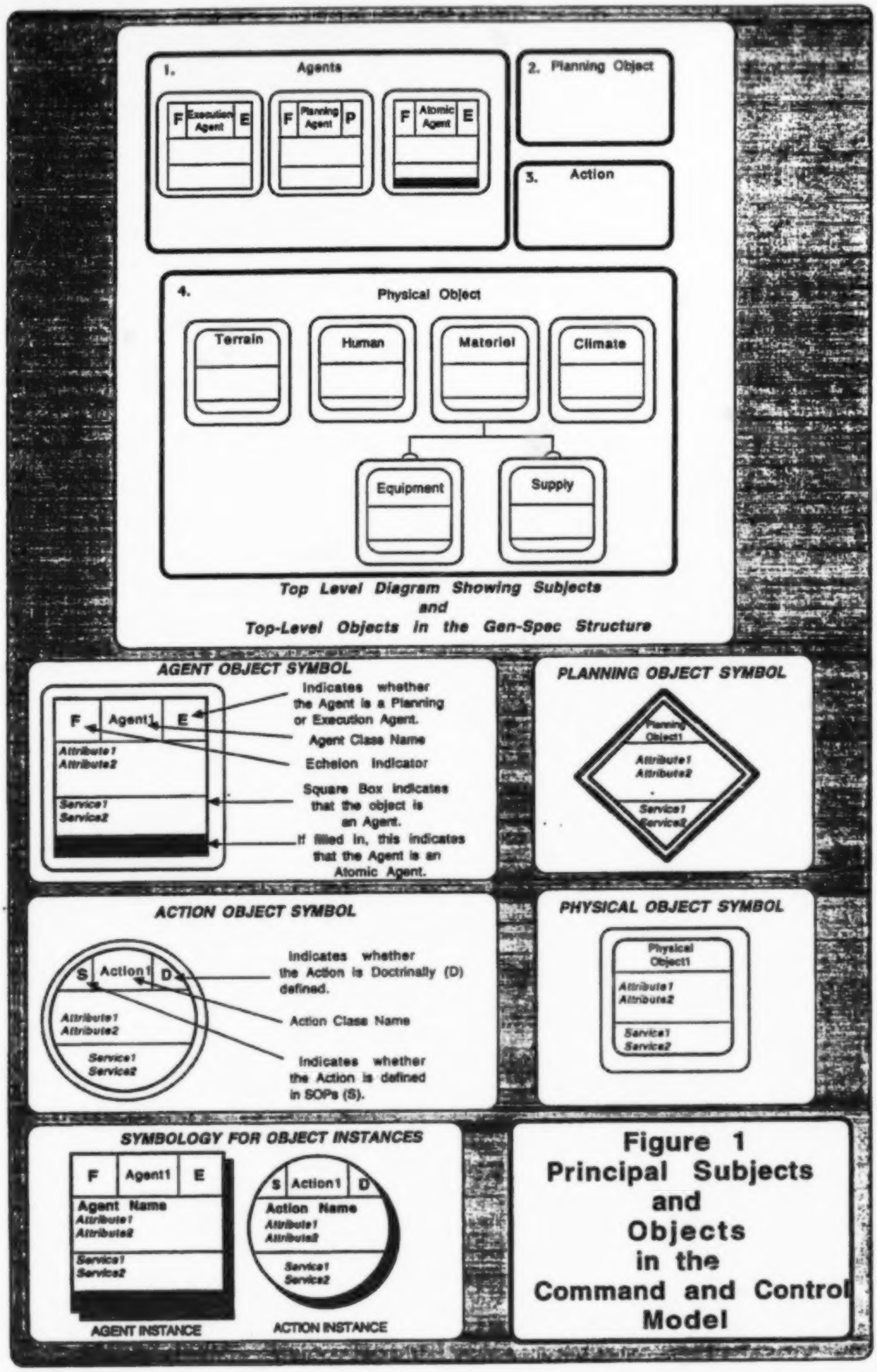

1. Agents
F Execution Agent E
F Planning Agent P
F Atomic Agent E
2. Planning Object
3. Action
4. Physical Object
Terrain
Human
Materiel
Climate
Equipment
Supply
Top Level Diagram Showing Subjects and Top-Level Objects in the Gen-Spec Structure
AGENT OBJECT SYMBOL
F
Agent1
E
Attribute1
Attribute2
Service1
Service2
Indicates whether the Agent is a Planning or Execution Agent.
Agent Class Name
Echelon Indicator
Square Box indicates that the object is an Agent.
If filled in, this indicates that the Agent is an Atomic Agent.
PLANNING OBJECT SYMBOL
Planning Object1
Attribute1
Attribute2
Service1
Service2
ACTION OBJECT SYMBOL
S
Action1
D
Attribute1
Attribute2
Service1
Service2
Indicates whether the Action is Doctrinally (D) defined.
Action Class Name
Indicates whether the Action is defined in SOPs (S).
PHYSICAL OBJECT SYMBOL
Physical Object1
Attribute1
Attribute2
Service1
Service2
SYMBOLOGY FOR OBJECT INSTANCES
F
Agent1
E
Agent Name
Attribute1
Attribute2
Service1
Service2
S
Action1
D
Action Name
Attribute1
Attribute2
Service1
Service2
AGENT INSTANCE
ACTION INSTANCE

Figure 1
Principal Subjects
and
Objects
in the
Command and Control
Model

mal set of actions required to carry out one or more actions based upon the action models. Stage and Phase objects are used to partition the elements of the Courses of Action, and eventually the Plan, into either independent or an ordered sequence of actions.

Summary

The use of these objects facilitates knowledge acquisition and permits the experts to rapidly validate the knowledge engineer's perception of the their knowledge. This Conceptual Model divides the Knowledge Base into Objects called Models which act as repositories of doctrine and expert knowledge. It uses Container Objects to coordinate, merge and modify the instances of the Model Objects to take into account contextual detail. Physical Objects are used to represent the Terrain, Climate, Human and Materiel found in the real world, whereas Agent Objects represent aggregations of Human and Materiel Objects. Planning Objects represent the abstractions (control measures) created to coordinate the execution of a Plan.

Automating the Plan Evaluation Process

The knowledge in a planning system is ultimately based upon answering the questions "Is this plan valid ?" and "Which one of these possible plans is better ?". Experts can normally characterize a good plan, but little of this expertise is documented. To automate the process one must understand the manual system, formalize the vocabulary used by the experts, acquire the knowledge in such a way that it can readily be extracted and used by both the system designer and validated by the expert, derive a metric with which to evaluate elements of a plan, formalize the criteria with which to rate a plan both absolutely and comparatively and finally have a consistent strategy with which to cope with recurring types of uncertainty.

Manual Plan Evaluation

Although there is no hard and fast way of evaluating a plan, there are certain guidelines which are espoused in the staff colleges. The planner first evaluates the assigned tasks/actions that he has to complete as well as the operational capability (as per Weisman 1992) of the agents he has available to effect these tasks. The commander then assesses, with respect to several standard factors such as terrain, the assigned actions and then derives several different outline plans, called friendly courses of actions, which permit him to carry out his assigned tasks. The most important factor on the battlefield is the enemy and, subsequently, his possible actions are analyzed and included in several possible enemy courses of action.

To evaluate the plan, the commander assesses each one of the friendly courses of action with respect to each one of the enemy courses of action as well as all of the other possible factors which could affect the plan's validity. The commander jots down the relative advantages and disadvantages of each one of the friendly courses with respect to each one of the enemy courses of action and then determines which one has the best chance of success. The Advantages and Disadvantages are based upon rules of thumb or heuristics, many of which are embedded in doctrine for application when relevant. The process is apparently simple, but some of the heuristics are based upon information which is extremely time--consuming to manually compile. For example the statement "If the ground on Approach A is passable to Enemy Tanks and there are no friendly anti-armour resources to cover the approach, then this constitutes a disadvantage" infers that the Intelligence personnel will first have to determine what types of armoured vehicle the enemy has and then the Terrain Analysis Team will have to carry out a specialized cross-country mobility analysis to see whether they can traverse the terrain on Approach A.

Formalization of the Vocabulary - Constraint Description

One of the first pre-requisites is the establishment of a clear-cut vocabulary, formalized in several distinct classes of constraints which can be represented using first order logic. The use of these well-defined constraints avoids confusion and ensures that what the expert says is accurately incorporated into the knowledge base. As per Weisman (1992) it was determined that for a military command and control system, there was a useful requirement for seven basic well-defined classes of constraints.

Instantiation constraints, as per the name, were used to permit the knowledge engineer to specify how to constrain the instantiation of the objects contained in the knowledge. Action Whole-part constraints contain the expert's knowledge concerning the decomposition of an action into sub-actions. Object Dependence constraints outlined the dependencies between Planning Objects and/or Actions. For example can you have an "Assembly Area" object without an associated action of "Move to Assembly Area" ? Plan Creation Constraints capture the logical connections in the planning process and answers the question of whether a plan is complete and, should the system be requested for assistance, which step in the planning process should be next. Spatial constraints capture the spatial relationships between the various planning objects. For example they would specify the exact meaning of "Above" and differentiate it from the term "Directly Above". Temporal constraints would specify whether certain actions have to be carried out BEFORE, AFTER, CONCURRENT or FIRM CONCURRENT with other actions. For example the Suppressing Fire action on the objective must be carried out concurrently (ie. FIRM CONCURRENT) with the Assault action. Finally Assignment constraints capture the formalities associated with the military command and control structure where certain agents are affiliated with other agents but only for very specific purposes. For example a commander can assign only certain actions to an agent which is under the command and control relationship "In Location" as compared to the relationship "In Support". All of these constraint categories are detailed in Weisman 1992, 1993a.

Factor Matrix

Once in possession of a set of mutually acceptable constraint terms, the knowledge engineer has to systematically determine what constitutes a good or bad plan. A Factor Matrix, as derived in Lindsay (1988) with the firm guidance of Brigadier General Kirby (ex-brigade commander and ex-commandant Canadian Land Forces Command and Staff College), is a knowledge acquisition tool specialized for the identification of the different heuristics associated with the evaluation of a plan. The matrix shows all of the probable plan elements (primarily actions and planning objects) listed against what have been determined to be the "Governing Factors" which correspond to the factors used in the military estimate process. For example the planning object "Assembly Area" is assessed against the factor "Enemy Air Capability" and the expert lists all objects, and their particular attributes, which the commander uses to determine the validity of his plan. What is now needed is a metric to measure the impact that a certain state, or combination of object attribute values, have on the assessment of a plan.

Metrics Used In Plan Evaluation

To simplify both the acquisition and expert validation of the plan evaluation knowledge, an Advantage-Disadvantage Metric was derived which directly corresponded to the military way of dealing with the uncertainty inherent with the heuristics contained in doctrine. A very favourable state is referred to as a Possible Critical Advantage and, conversely, a very unfavourable state as a Possible Critical Disadvantage. This metric includes a range of intermediate values which are defined in detail in Weisman 1992. However this metric was found to be inadequate as certain factors were either more or less relevant in certain situations and that is why the adjective "Possible" was added to each assessment.

To handle the inadequacy, a manner of assigning weights, the Factor Importance Metric, was introduced which permitted the expert to state general heuristics (eg. Maintain a certain inter-vehicle distance to minimize the effects of an air attack) using the Advantage-Disadvantage metric and then permit him to separately express the relevancy of that general rule using the Factor Importance Metric (eg. If the Enemy Air is ineffective/non-existent then the factor is irrelevant). This flexibility permitted the expert to describe several states/rules that would generally lead to a possible advantage or disadvantage and then qualify the assessment with one or two rules which would determine the actual relevance of the factor itself. This metric classified factors in the range of Critical to Irrelevant.

The Plan Evaluation Metric, as illustrated in Figure 2, was derived to combine the Advantage--Disadvantage assessment with the Factor Importance to definitively determine whether a plan had a specific advantage or disadvantage. This metric identified elements of a plan as being either Critically Advantageous or Critically Disadvantageous, inferring a clear-cut advantage / disadvantage with virtually no chance of failure / success, Advantageous or Disadvantageous, in that there is a clear advantage / disadvantage with little chance of plan failure / success, or Neutral, which confers neither an advantage or disadvantage. There is also a Possible Resource Reallocation State which identifies a state where a significant advantage exists for a factor which is not very important. This implies that there may resources not being allocated to the places where they could be best contribute to a plan's success. For example if significant, and always scarce, resources are being tasked with the protection of a convoy against air attack, it may result in a Possible Critical Advantage. However if the factor Enemy Air Capability is Irrelevant, then one could infer that those resources could be better used elsewhere in, say, a defensive role.

Now the Plan Evaluation Metric satisfies the requirement to critique one specific aspect of the plan, now one must consider how the overall plan can be critiqued and how the plan can be comparatively assessed with respect to other candidate plans.

Plan Evaluation

Based upon the heuristics used by military commanders, two Plan Evaluation Criteria were derived, namely the Least Risk and Best Advantage Criterion. The following will discuss these two concepts in more detail.

The Least Risk Criteria takes all of the advantages and disadvantages associated with a plan and classifies the entire course of action with respect to the value of the least advantageous rating. Therefore a plan with two Critically Advantageous States, four Advantageous States and one Disadvantageous State will be classified as a Disadvantageous Plan. This assessment is analogous to describing the strength of a structure/ chain as a function of its weakest link / component. The advantage of this method is that it isolates the plan weakness(es) but its major shortcoming is that it does not place those) weakness(es) in perspective. For example a plan may have ten Critically Advantageous States and one Disadvantageous State yet the entire plan is classified as Disadvantageous.

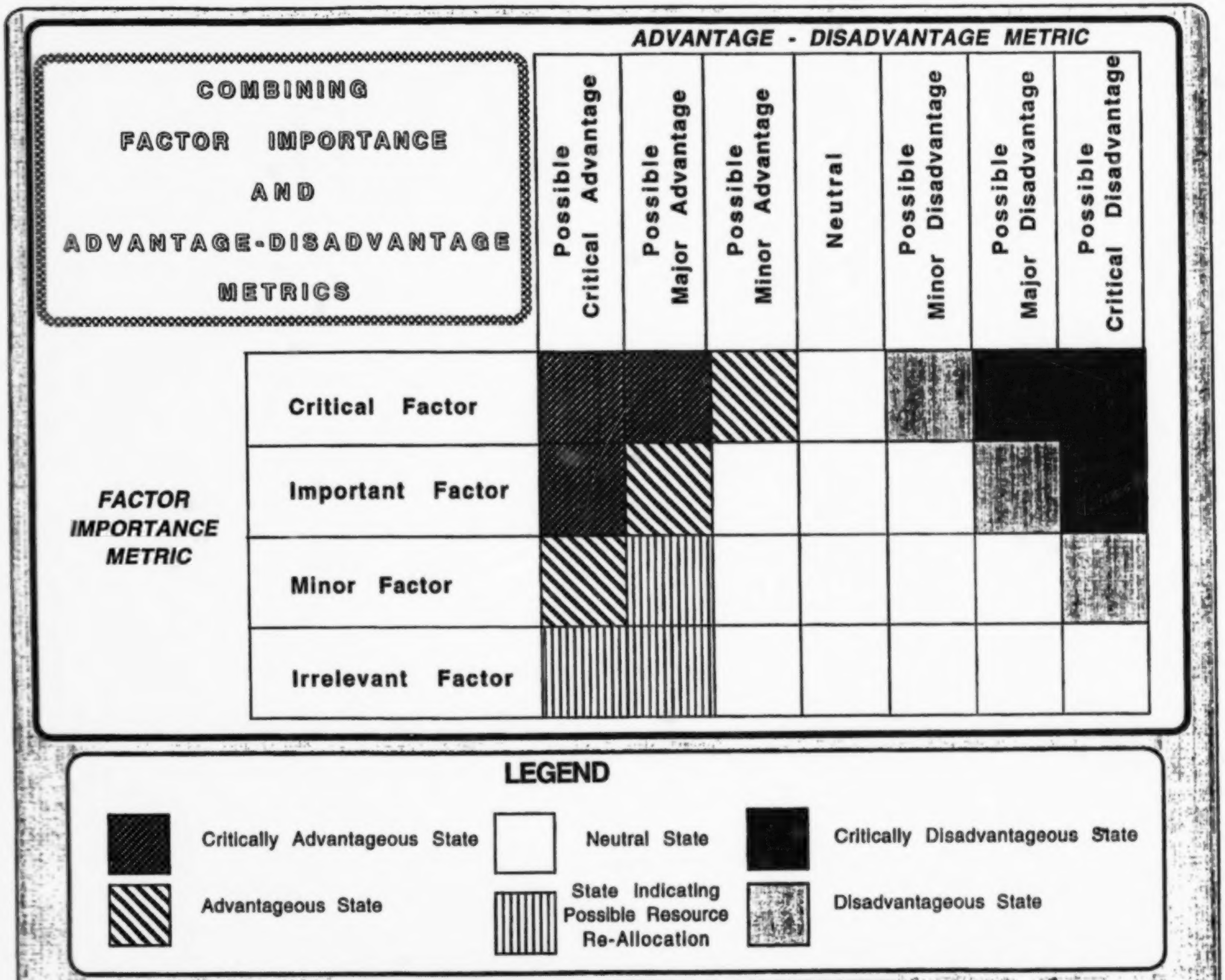
The other way to assess a plan is with respect to the Best Advantage Criterion, which is basically a weighted average. This evaluation accepts the fact that there may be a fair degree of risk involved but the critical advantages may outweigh the critical disadvantages. One way of doing this is to use a weighted sum with the states being represented by the following weights:

a.	Critical Advantage (CA)	+10
b.	Advantage (A)	+1
c.	Neutral (N)	0
d.	Disadvantage (D)	-1
e.	Critical Disadvantage (CD)	-10

The weights are heuristically determined, with the skewing meant to highlight the Critical Advantages and Critical Disadvantages. For example, considering the previous situation with ten Critical Advantages and one Disadvantage, this assessment would give the plan a score of 99. The clear danger with this assessment is that it can obscure a critical disadvantage which could fatally flaw the plan. On the other hand it places the overall advantage of the plan in perspective.

Plans can be relatively compared using both of the above criteria, with the Least Risk criteria isolating what the system perceives to be the weakest part of the plan and the Best Advantage Criteria identifying the plan with the most advantages. Figure 3 illustrates the matrix of friendly courses of action being assessed relative to the enemy courses of action. The system assesses and presents the evaluation in accordance with the two criteria, in the same manner as in the manual system.

One can also assess the Overall Validity of the planning process and determine, by inspection, whether one is caught in a dilemma. Examining Figure 3, one sees that there is no course of action which does not possess at least one Disadvantage as indicated by the Least Risk criteria. Therefore the best plan will still be flawed. At this point the planner may accept the fact that there exists only a set of unpleasant options, or he may decide to recommence the planning process and create another course of action which may address some of the deficiencies of the previously derived courses of action.



COMBINING FACTOR IMPORTANCE AND ADVANTAGE-DISADVANTAGE METRICS

FACTOR IMPORTANCE METRIC	Possible Critical Advantage	Possible Major Advantage	Possible Minor Advantage	Neutral	Possible Minor Disadvantage	Possible Major Disadvantage	Possible Critical Disadvantage
Critical Factor	▩	▩	▧	☐	▒	■	■
Important Factor	▩	▧	☐	☐	☐	▒	■
Minor Factor	▧	▥	☐	☐	☐	☐	▒
Irrelevant Factor	▥	▥	☐	☐	☐	☐	☐

ADVANTAGE - DISADVANTAGE METRIC

LEGEND

- ▩ Critically Advantageous State
- ▧ Advantageous State
- ☐ Neutral State
- ▥ State Indicating Possible Resource Re-Allocation
- ■ Critically Disadvantageous State
- ▒ Disadvantageous State

Figure 2 - Plan Evaluation Metric

Dealing with Certain Forms of Uncertainty

There are two basic forms of uncertainty normally encountered in plan evaluation. First the set of possible values (domain) for an attribute may not be sufficiently precise and more than permissible attribute value can represent the actual state. Although this should have been detected by the knowledge engineer in the initial analysis, unforeseen things happen on operations. For example if the set of possible values for Climate.Precipitation is restricted to {none, rain, snow, hail} and it is actually "sleet" then both the values of rain and snow could be applicable. Secondly there is the case where it is known that the actual value of the object attribute is restricted to one element of a subset of the domain. For example, the meteorologist may realize that for an upcoming operation that there is going to be precipitation, but he is not sure whether it will be rain or snow.

On the battlefield the consequence of error is significant and commanders normally plan for the worst case scenario. In accordance with this practice, one deals with uncertainty by assessing plan elements using ALL of the likely values for the attribute and selecting, for the final evaluation, the most disadvantageous value. For example if the value of "Snow" results in an Advantageous state and the value "Rain" results in a Critically Disadvantageous state, then the value "Rain" will used for the purposes of plan evaluation.

Discussion

The evaluation of plans is actually an ongoing process. This is due to the fact that a lengthy period of time elapses between the derivation and actual implementation of a plan . Therefore some of the facts, or Object Attribute Values, upon which the plan is based, may well change and thereby alter the previous plan evaluation. Due to structure of the knowledge in the conceptual model and the initial plan evaluation, the system knows what Object Attribute Values have to be monitored and whenever one of those values either changes (eg. It suddenly starts to rain) or does not change according to plan (eg. Unit A is not at Point B at Time C), the system can restart the plan evaluation process and notify the duty staff and commander of any potential problems. In other words this metric provides the basis for not only deriving a plan but also for monitoring the continued validity of the plan. This paper has only cursorily dealt with uncertainty or default reasoning, however the problems associated with these issues are covered in Weisman 1992 and 1993b.

Implementation and Prototyping

Selected parts of the overall model have been prototyped using Expert system tools on Personal Computers, using the VPExpert shell, and on Unix workstations, using the Automated Reasoning Tool (ART). The ART tool was used to both manage a very large and complex agent (1st Canadian Division) as well as to assess a small plan, based on the plan model for a tactical road move. The power of the ART Inference engine became evident in that it took hundreds of rules in VP-Expert to achieve the evaluation whereas a few well-formulated rules would suffice in ART. Notably ART was able to easily check for both the presence of incorrect information as well as the absence of certain critical information in the knowledge base. Although ART is far more difficult to use, it, or a similar product, is superior for the development of complex systems.

The prototypes have helped in the derivation of the conceptual model and, conversely, there have been few difficulties testing the various elements of the conceptual model indicating that it is encouragingly robust. A key advantage of the overall model is its versatility in that it is applicable to a wide range of low, medium and high intensity operations. For example it is presently being used to model emergency response to a major oil spill as discussed in Weisman 1993a. Another aspect, and possibly the most important one, is that the model provides a consistent framework for the sharing of the acquired knowledge which is machine/system independent. Even if the prototype code is thrown away, this expert knowledge, which is difficult to acquire and is actually the core of the system, is still structured in a standard way and available for use elsewhere.

In conclusion, there is still a tremendous amount of work to be done, however the derivation of a conceptual model, including this simple yet effective means of evaluating plans, provides a solid foundation for further research. There is little doubt in the author's mind that an intelligent

	ENEMY Course of Action 1		ENEMY Course of Action 2		ENEMY Course of Action 3	
FRIENDLY Course of Action 1	11 A ASSEMBLY AREA PROTECTED A ATTACK POSITION PROTECTED D ENEMY ON OBJECTIVE D ENEMY DEFILADE OBJECTIVE		12 CD AXIS OF ASSAULT IN DEFILADE A ASSEMBLY AREA PROTECTED A ATTACK POSITION PROTECTED D ENEMY DEFILADE OBJECTIVE		13 CA AXIS OF ATTACK BEHIND ENEMY A ASSEMBLY AREA PROTECTED A ATTACK POSITION PROTECTED	
	LEAST RISK	*BEST ADVANTAGE*	*LEAST RISK*	*BEST ADVANTAGE*	*LEAST RISK*	*BEST ADVANTAGE*
	Disadvantage	1 + 1 -1 -1 0	Critical Disadvantage	-10 + 1 +1 -1 -9	Advantage	+10 + 1 +1 22
FRIENDLY Course of Action 2	21 A ASSEMBLY AREA PROTECTED A ATTACK POSITION PROTECTED D ENEMY ON OBJECTIVE CD AXIS OF ASSAULT FULL VIEW OF ENEMY		22 A ASSEMBLY AREA PROTECTED A ATTACK POSITION PROTECTED D ENEMY ON OBJECTIVE CD AXIS OF ASSAULT FULL VIEW OF ENEMY CD AXIS OF ASSAULT IN RANGE ALL ENEMY FORCE		23 A ASSEMBLY AREA PROTECTED D ATTACK POSITION IN VIEW OF ENEMY CD ATTACK POSITION IN DF RANGE OF ENEMY CD AXIS OF ASSAULT FULL VIEW OF ENEMY CD AXIS OF ASSAULT IN RANGE ALL ENEMY FORCE	
	LEAST RISK	*BEST ADVANTAGE*	*LEAST RISK*	*BEST ADVANTAGE*	*LEAST RISK*	*BEST ADVANTAGE*
	Critical Disadvantage	1 + 1 -1 -10 -9	Critical Disadvantage	1 + 1 - 1 -10 -10 -19	Critical Disadvantage	1 - 1 -10 -10 - 10 -30
FRIENDLY Course of Action 3	31 A ASSEMBLY AREA PROTECTED A ATTACK POSITION PROTECTED D ENEMY ON OBJECTIVE		32 A ASSEMBLY AREA PROTECTED A ATTACK POSITION PROTECTED		33 A ASSEMBLY AREA PROTECTED D ATTACK POSITION IN VIEW OF ENEMY CD ATTACK POSITION IN DF RANGE OF ENEMY CD ENEMY ON AXIS OF ASSAULT BEFORE OBJ CD AXIS OF ASSAULT FULL VIEW OF ENEMY CD AXIS OF ASSAULT IN RANGE ALL ENEMY FO	
	LEAST RISK	*BEST ADVANTAGE*	*LEAST RISK*	*BEST ADVANTAGE*	*LEAST RISK*	*BEST ADVANTAGE*
	Disadvantage	1 + 1 -1 1	Advantage	1 + 1 2	Critical Disadvantage	1 -1 -10 -10 -10 -10 -40

EVALUATION SUMMARY

	LEAST RISK			Evaluation	BEST ADVANTAGE			Evaluation
Friendly Course of Action 1	D	D	A =	D	0	-9	+12 =	+3
Friendly Course of Action 2	CD	CD	CD =	CD	-9	-19	-30 =	-58
Friendly Course of Action 3	D	A	CD =	CD	+1	+2	-40 =	-37
OVERALL PLAN				D				

Figure 3 - COMPARISON OF OPTIONS AND EVALUATION

decision support system could provide valuable assistance to, initially, students at the staff colleges, and eventually to battlefield commanders.

References

1. Coad, Peter; Yourdon, Edward, 1991a: "Object-Oriented Analysis" (Second Edition) Prentice-Hall Inc. Englewood Cliffs, New Jersey 1991.

2. Coad, Peter; Yourdon, Edward, 1991b: "Object-Oriented Design" Prentice-Hall Inc. Englewood Cliffs, New Jersey 1991.

3. Lindsay, E., Major 1989: "A Methodology For Knowledge Acquisition Used in The Design of a Decision Support System", Master's Thesis, Royal Military College, March 1989.

4. Tactical Command, Control and Communications System (TCCCS) Project, 1989 "Automated Combat Information System Conceptual Process Model (ACIS CPM)", Contract W8476-6-EG04/01NJ dated 13 January 1989.

5. Tactical Command, Control and Communications System (TCCCS) Project, 1988b "Conceptual Data Model for the Army Command Information System" unpublished draft, dated 8 August 1988.

6. Weisman, Robert; Fugere, Jean et al 1990a: " A Conceptual Model For Command and Control" Proceedings of the 2nd Symposium on Applications of Expert Systems in DND (Department of National Defence) held at the Royal Military College of Canada, Kingston, Ontario, May 1990.

7. Weisman, Robert; Fugere, Jean et al 1990b: " A Knowledge-Based Approach to Military Planning", Canadian Defence Science Symposium, c/o Chief Research and Development, National Defence Headquarters 101 Colonel By Drive, Ottawa, Ontario 1990.

8. Weisman, Robert 1992:" A Conceptual Model for Military Command and Control" MSc Thesis, Department of Mathematics and Computer Science, Royal Military College of Canada, Kingston, Ontario, K7K 5LO.

9. Weisman, Robert 1993: " Modelling Command and Control for Emergency Response Scenarios: A Marine Oil Spills Perspective", Arctic and Marine Oil Spill Technical Seminar, held in Calgary 7-10 June 1993, sponsored by Environment Canada.

10. Weisman, Robert 1993b:"Dealing With Uncertainty In Automated Decision Support Systems For Emergency Response Scenarios", Proceedings sponsored by Major Industrial Accidents Council of Canada and Environment Canada held in St. John, New Brunswick 7-10 September 1993.

909945

DND Knowledge-based Systems and Robotics Workshop, Nobember 1993

Decision Aids for Intelligence Fusion

Marlène Gauvin[1]
DREV[2] Command and Control Division
2459 Blvd. Pie XI north,Courcelette, Quebec, GOA 1R0

Michel Lizotte[3]
DMR Group Inc.
Pl. Iberville III, 400-2960 Blvd. Laurier, Sainte-Foy, Quebec, G1V 4S1

Dominic Roy[4]
DREV Command and Control Division
2459 Blvd. Pie XI north,Courcelette, Quebec, GOA 1R0

#149663

Abstract

We have undertaken a five year research project to explore and provide solutions, in terms of smart computing assistance, to intelligence fusion: the process of combining battlefield information to achieve the most accurate understanding of the enemy situation.

The military expertise involved in this process is so complex that any intelligent component which attempts to emulate part of it would not be accepted by intelligence analysts without a long process of confidence building in the effective capacity of such decision aids. As such, a two-stage approach has been adopted; it consists of first building a powerful fusion man-machine interface and subsequently expanding it to incorporate more sophisticated automated facilities. We define a central concept, called hypothesis, to represent the elementary assumptions considered in the analysis process. User and machine manipulation of hypotheses are respectively considered in the first and second stage of the approach. Manipulations are essentially performed through a defined set of specialized fusion operations, such as tracking and aggregating, which model the analysis process.

The first stage is now complete. We have built a Tactical Information Fusion prototype that allows an assessment of the enemy situation to be performed using cooperative analysis from several intelligence workstations. Each workstation provides a Graphical Reasoning Support System that allows users to visualize the manipulation of hypotheses during fusion operations. These concepts were validated and proved through an abbreviated command post exercise. The need for advanced automation was recognized by the users who encouraged us to proceed to the second stage. We propose a method to gradually incorporate decision aids. Some specific research areas that could benefit from automation have been identified. Issues involved in designing an architecture to synchronize decision aids are also discussed.

Introduction

The fusion of intelligence which is aimed at understanding the enemy situation, in terms of its position, strength and organization, is a critical phase in the planning and conducting of military operations for all levels of intensity conflicts. The timeliness and the accurateness of the enemy picture must be maximized to insure judicious and effective decision-making by Commanders. The need for effective *and now smart* computing support in the assessment of the enemy situation is generally recognized in the command and control intelligence community; on the one hand, intelligence analysts are overloaded with large flows of information from modern surveillance systems and troops in contact, and on the other hand, they have to cope with the fuzzy character of the information, such as inaccuracy, incompleteness, redundancy and even, incoherence. These two problem dimensions gave us the primary motivation for our research work. To this end, we started four years ago a major research project to explore and provide an advanced computing environment, with appropriate decision aids, to help intelligence analysts in the integration of the various information (Ref. 1). This project contains several axes of efforts, such as understanding the process of intelligence fusion in the global context of military operations, examining the applicability of the various state-of-the-art information technologies, conceiving and building a computing environment appropriate for team work analysis, defining command post exercises for validation purposes and providing suitable decision aids. This latter aspect is discussed in detail in this paper.

---

[1] Phone (418) 844-4412

[2] DREV = Defence Research Establishment Valcartier

[3] Phone (418) 653-6881

[4] Phone (418) 844-4438

We present an incremental approach consisting of two stages to provide decision aids to the private work of intelligence analysts. While knowing that intelligence fusion is performed by a team of several analysts, often organized in a hierarchical manner, we concentrate on the individual work that consists in developing and maintaining a personal perception of the evolving enemy situation. Decision aids to support the integration of the different views are not covered in this paper.

The Two-Stage Approach

Several factors must be taken into account when decision aids are required in a working environment where the task in itself is complex and performed in a work-loaded and stress context. In particular, it is of prime importance to develop an adequate model of the decision work and to ensure that users are willing to develop a solid thrust towards the capabilities of the future decision aids. With those two concerns in mind, it becomes easier to build a system that will fulfill their requirements while increasing their productivity. Of course, such a system will have to be intuitive and user-friendly and achieve real-time responses.

These principles are applicable to the present case; intelligence fusion is performed in a very difficult context. After having approached intelligence military experts, we were confronted with their apprehension towards automation. The intelligence work involves different areas of specialized expertise that are highly integrated or intimately linked. The domain knowledge is so complex that it became obvious that any intelligent component which attempts to emulate part of their work would not be accepted without a long process of confidence building. This suggested the approach towards the delivery of decision aids adapted to the intelligence fusion work (Ref. 2).

At first sight, the approach is simple, consisting of two stages. The first stage is aimed at specifying the basis for decision aids by defining a model of the fusion work and validating it through experimentation with a powerful man-machine interface. A prototyping development method is suitable to increase interactions between intelligence analysts and system engineers in order to come up with a man-machine interface tailored to the fusion work. The second and last stage is aimed at incorporating more automated facilities, including advanced decision aids.

Stage 1 - Basis Decision Aids

Fusion Work Modeling

As in any decision aids system, a modeling phase of the domain knowledge is necessary. It consists of specifying a conceptual framework by attempting to identify the basic concepts used in the reasoning (also called reasoning objects) and to draw out the elementary operations performed to achieve the work goals. The acquisition of knowledge to develop the conceptual framework of the intelligence fusion work was provided from three important sources of information: (1) frequent interviews with intelligence experts, (2) military documentation that describes the production of intelligence (Refs. 3, 4 and 5) and, (3) other fusion work described in open literature (Refs. 6, 7 and 8).

After examining the fusion process more closely, we identified four basic concepts that intelligence analysts have to deal with in the course of their work. They are:

- *Information:* In the intelligence fusion context, it is any element of a military message that reports on the current enemy and environmental conditions. A typical example of information would be a sighting of a motor rifle regiment at a specific sector on the battlefield. There are many types of information, such as enemy sightings, terrain descriptors and enemy assessments from other intelligence cells. But in this paper we will only consider the main types which are units and equipment, the entities of the battlefield.
- *Hypothesis:* It is an elementary assumption on any information considered by the analyst in the course of his work, whether it originates from a message or is produced interactively during fusion operations. The term hypothesis has been chosen to emphasize the degree of uncertainty associated with it. The hypothesis constitutes the central concept of the fusion work.
- *Cluster:* It expresses different relationships between hypotheses and is formed during a fusion operation. The intelligence fusion process is aimed at repeatedly clustering hypotheses in order to draw out clearer and more coherent pictures of the battlefield. The subordination between two unit hypotheses is a kind of relationships that could be expressed by a cluster. A cluster is also an assumption on the elements constituting the hypothetical enemy picture.
- *Situation:* It represents the whole enemy picture as perceived by the intelligence analyst. It is expressed as a set of hypotheses

and clusters. In the context of the fusion work , the situation is always hypothetical.

Figure 1 shows these concepts in a semantic network to highlight their interrelationships.

We defined a set of five fusion operations that cover the main issues of intelligence fusion and help express the analyst's understanding of the tactical situation. They operate on hypotheses and result in a cluster. They are: (1) *tracking*: to indicate that hypotheses represent the same enemy entity at different times; (2) *Merging*: to indicate that multiple hypotheses refer to the same entity on the battlefield; (3) *Aggregating*: to indicate that a group of units belongs to the same higher formation; (4) *Allocating*: to allocate a given unit to a temporary formation or to express that a piece of equipment belongs to a specific unit and (5) *Relating*: to express an arbitrary relationship between hypotheses. We also considered in our study the complementary operations to achieve intelligence fusion, such as the capability to create, modify and delete hypotheses, to undo a fusion operation, and to have access to external intelligence information.

Computing Environment

The work model has been implemented as part of a larger computing environment, referred to as the Tactical Information Fusion (TIF) Prototype, that allows a cooperative assessment of the enemy situation to be performed from several intelligence analysis workstations. The TIF prototype provides sets of specialized functions to the different people involved in intelligence processing, such as the logger who receives and annotates the external message traffic, the various intelligence analysts and the senior analyst who integrates partial analysis to produce higher level intelligence assessment.

The prototype also provides a relational data base server to make accessible other knowledge of the intelligence domain analysts may required in their fusion work. This knowledge includes doctrinal information, such as the enemy organizational structure, signature and deployment templates, and order of battle information (Ref. 9).

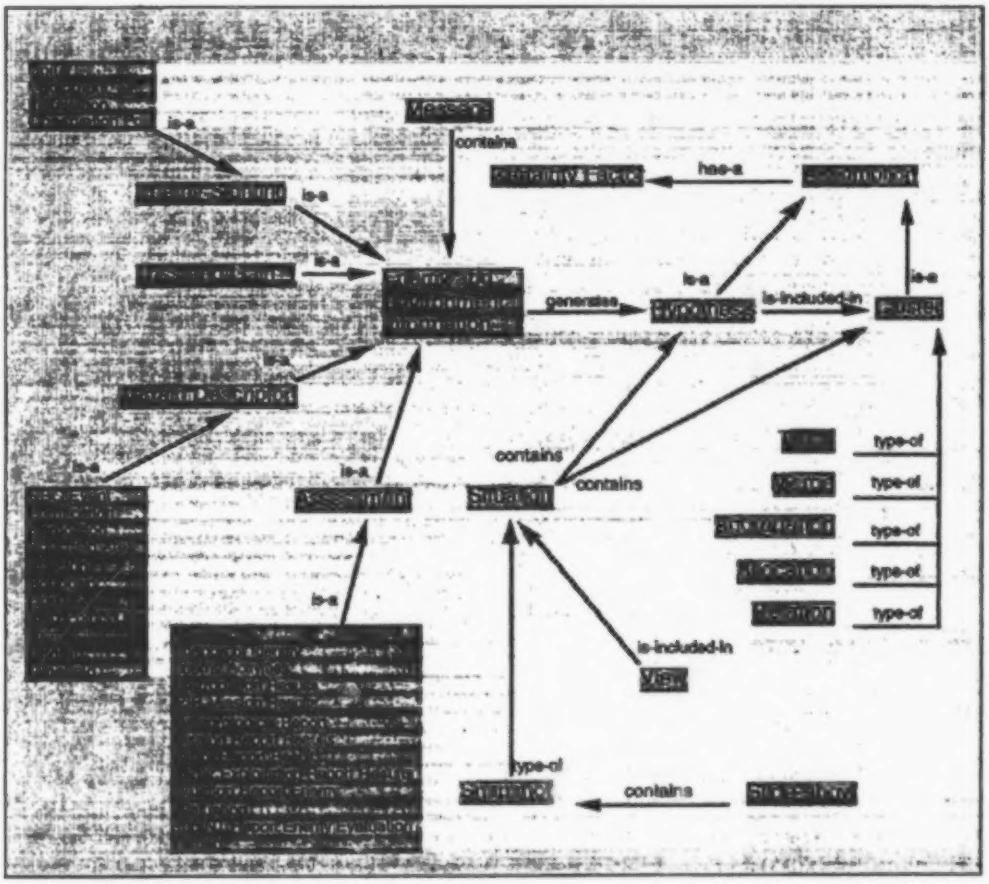


FIGURE 1: Semantic Network of Intelligence Fusion Basic Concepts

The prototype relies on a technological architecture that allows a harmonious combination of various state-of-the-art computing technologies, such as object-oriented approach, geomatics, graphical user interfaces, relational databases, distributed processing and simulation facilities. The modularity of the architecture allows gradual addition of technological components, such as workstations and servers, and facilitates integration of emerging technologies, such as artificial intelligence (AI) techniques, multimedia and visualization. The man-machine interface of analysis workstations (Ref. 10) combines a multi-window environment, a geographic mapping interaction facility supporting the display of multiple views of the situation over a map background and a Graphical Reasoning Support System (GRSS) to visualize analyst's manipulations of hypotheses during fusion operations.

Graphical Reasoning Support System

The GRSS provides the basis decision aids to the intelligence fusion work by supporting fusion operations through graphical interactions and by making explicit, upon request, the chain of reasoning that has been followed during fusion operations.

The analyst's working space (hypothetical situation) can be rapidly crowded with hypotheses. By continuously combining hypotheses and considering newly information, the analyst creates other hypotheses. He may end up with a fairly complicated *network of transactions*, where each transaction is very likely the fruit of a reasoning decision. The GRSS allows the network to be consulted so as to provide explanations on how hypotheses were derived. It indicates whether a particular hypothesis originates from a message, constitutes a new version of another hypothesis, was issued by the analyst or was produced by a fusion operation.

In the last case, the cluster in which belongs the hypothesis is made explicit; it is graphically displayed and/or textually described in a window. In the graphic mode, interrelations between hypotheses are displayed as lines joining their respective military symbols over the map background.

In the same way as it supports the fusion operations, the GRSS also allows analysts to backtrack in their reasoning and undo operations. It keeps track of any hypotheses manipulation, allowing in this way a previous hypothetical situation to be retrieved. Towards this end, hypotheses are associated with an activation status to indicate if they currently contribute to the situation or not. Inactive hypotheses are considered obsolete and are no longer required to represent the tactical situation. They can only be consulted or reactivated; no other operation is allowed. Hypotheses are also associated with a visibility status to indicate if they are displayed on the working map. Both status are used to determine the operations allowed. They can be modified by the analyst or altered upon the use of a fusion operation.

In addition to fusion operations and reasoning aids, the GRSS provides briefing aids to report on the evolution of the situation. Images of the situation prevailing at different times can be stored in snapshots and displayed dynamically with a slide-show facility.

The following subsections illustrate how the GRSS supports the fusion operations through the graphical interactions.

Tracking operation

This operation is used to indicate that some unit hypotheses $U_1, U_2, ..., U_n$ represent the same moving enemy entity at different times. The U_i representing the most recent one in the cluster stays active while the others are inactivated and hidden. On request, the system displays on the map a line joining all locations of inactivated hypotheses. Figures 2 and 3 illustrate a tracking operation.

The GRSS also supports the reasoning of the analyst by issuing a warning when such an operation is applied on an aggregated or allocated unit hypothesis. This way the analyst is made aware that clusters resulting from an aggregating or allocating operations can become obsolete and may have to be updated to be consistent with the current operation. In other words, the analyst has identified a movement of a unit which should normally involve similar movement of the previously clustered hypotheses.

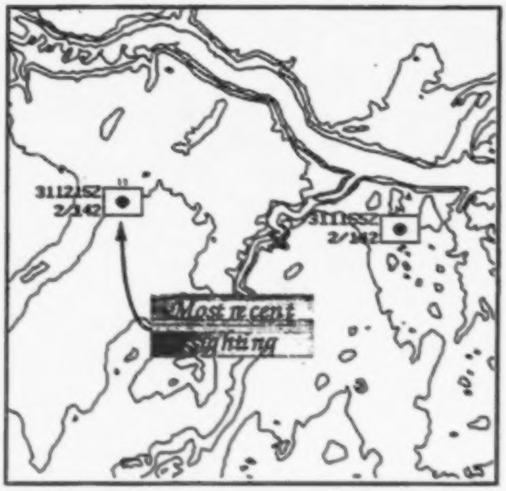


FIGURE 2: Before Tracking

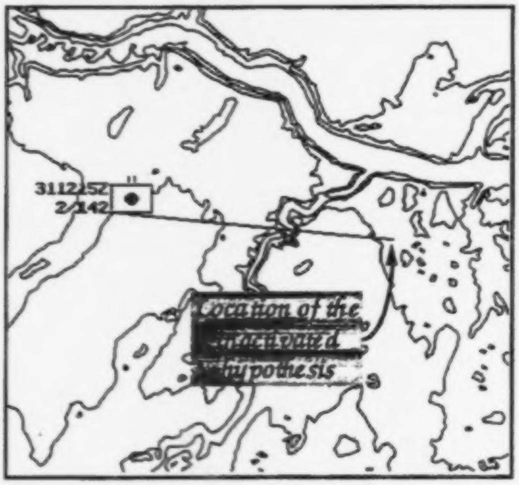

3112152Z
2/142
Location of the inactivated hypothesis

FIGURE 3: After Tracking

Merging operation

This operation is used to indicate that some unit sightings $U_1, U_2, ..., U_n$ observed around the same time and location refer to the same entity (a gap between locations may indicate erroneous location values).

A new unit hypothesis U that combines attribute values of all clustered hypotheses U_i is automatically produced and positioned at the center of those U_i. The time of sighting of U is set to the most recent one of U_i. The merging of attribute values other than the location and the time of sighting is performed in a straightforward manner. Merged hypotheses are inactivated and hidden. On request, the system displays the clustered hypotheses. Figures 4 and 5 illustrate a merging operation.

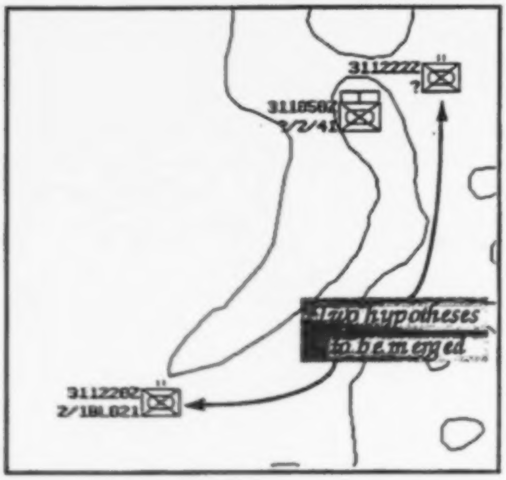

3112222Z
?
3110502Z
2/2/41
Two hypotheses to be merged
3112202Z
2/1BL821

FIGURE 4: Before Merging

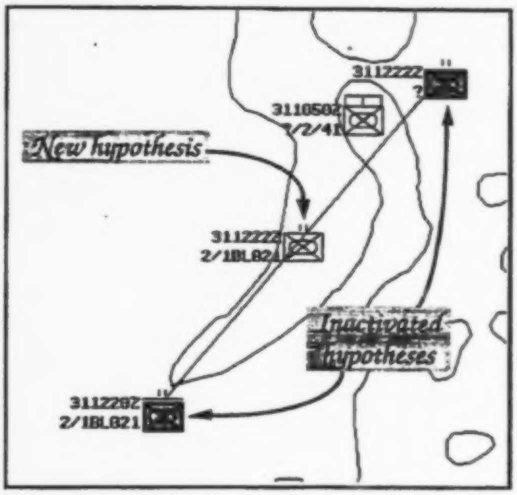

3112222Z
?
3110502Z
2/2/41
New hypothesis
3112222Z
2/1BL82
Inactivated hypotheses
3112202Z
2/1BL821

FIGURE 5: After Merging

In the same way as within the track operation, a warning is issued when a merging operation is applied on an aggregated or allocated hypothesis.

The deletion of a merge cluster inactivates the hypothesis U produced when the cluster was built. Before the deletion of a cluster, a check is done to verify if U belongs to other clusters. If so, the system warns the analyst of a such situation and will not delete the cluster. This prevents the analyst from undoing a merge if its resulting hypothesis has been used as input within another fusion operations.

Figure 6 shows a chain of reasoning where a merge cluster cannot be deleted because the resulting hypothesis has been allocated. In this model, hypotheses and fusion operators are respectively shown in circles and rectangles.

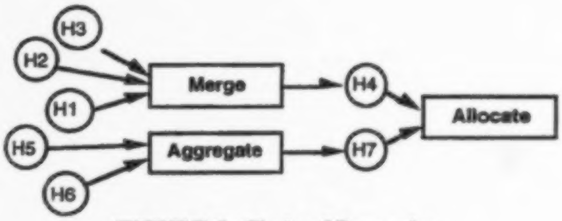

H3
H2
H1
H5
H6
Merge
Aggregate
H4
H7
Allocate

FIGURE 6: Chain of Reasoning

Allocating operation

This operation is used to allocate units $U_1, U_2, ..., U_n$ to a temporary formation U. If the analyst applies again this operation to allocate other units to U, the system will not create a new cluster but will automatically add the new allocated units to the cluster. All allocated units are inactivated and hidden. On request, the system displays the clustered hypotheses either graphically (Fig. 7) or textually (Fig. 8).

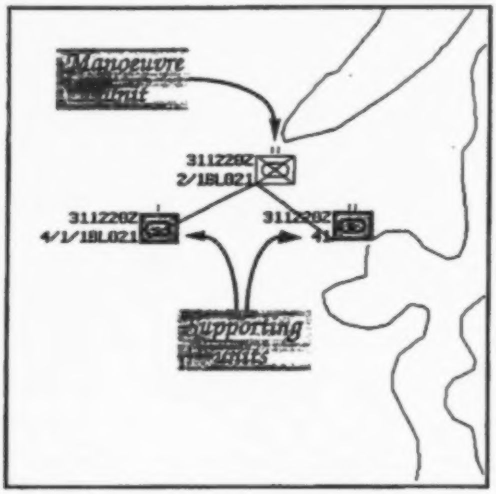
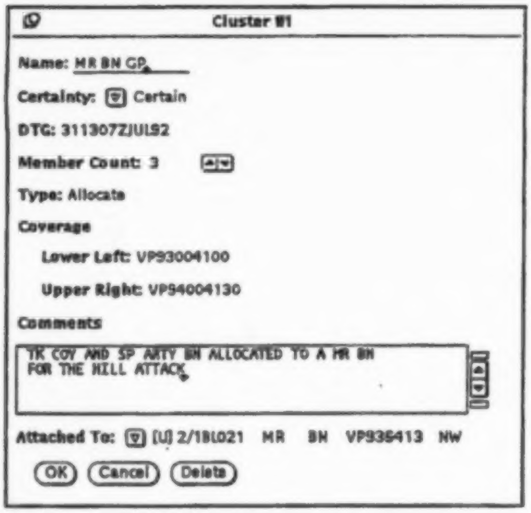


FIGURE 7: A Graphical Allocate Cluster



Name: MR BN GP

Certainty: Certain

DTG: 311307ZJUL92

Member Count: 3

Type: Allocate

Coverage

Lower Left: VP93004100

Upper Right: VP94004130

Comments

TK COY AND SP ARTY BN ALLOCATED TO A MR BN
FOR THE HILL ATTACK

Attached To: [U] 2/1BL021 MR BN VP936413 NW

FIGURE 8: An Allocate Cluster Window

A hypothesis U_i can be allocated to only one unit at a time. When the allocating operation is applied on an allocated hypothesis, the system supports the analyst's reasoning by warning him the operation will result in the deletion of the existing cluster. The analyst can then resume or cancel the current operation.

Aggregating operation

This operation is used to indicate that some unit hypotheses $U_1, U_2, ..., U_n$ belong to the same higher formation U. A blank unit hypothesis U is automatically produced and positioned at the center of the clustered hypotheses. This operation also sets the parent unit attribute of the aggregated units. When it is applied on an aggregated unit, the analyst is warned that the operation will result in the deletion of the existing cluster. The analyst can then resume or cancel the operation.

In the same way as within the merging operation, the deletion of an aggregate cluster is not performed before verifying if the parent hypothesis was used in another operation.

Relating operation

This operation is used to express and annotate a certain relationship between some unit hypotheses $U_1, U_2, ..., U_n$. A cluster created this way can be subsequently used as input to any other fusion operations.

Validation

The basis decision aids provided by the GRSS were validated through an abbreviated command post exercise wherein six military intelligence specialists used the prototype workstations to integrate the tactical information coming from a simulated battlefield scenario (80 messages per hour). The aim was to evaluate if the conceptual framework was representative of the intelligence fusion work. The result was totally positive. Experts felt comfortable with the concepts, and particularly with the fusion operations, which were found relevant and useful to the fusion work. The man-machine interface was also highly rated. After a very short training period, intelligence experts rapidly became autonomous and efficient in using these decision aids. Results were collected during a debriefing session and through a six-page questionnaire; their assessment will be published later.

Stage 2 - Advanced Decision Aids

This stage focuses on the gradual incorporation of more sophisticated automated facilities to provide an intelligent form of fusion assistance to the intelligence analyst. The objective is to develop a set of advanced decision aids which will work autonomously to the resolution of certain problems whose scope and complexity will have been delimited by the analyst. These aids should adhere to the conceptual framework which was worked out and validated during the preceding stage. The analyst will be encouraged to use them. He will have the ability to endorse or ignore the conclusions reached by the aids and to turn them on and off as required. An aid could be used, for instance, to detect newly arrived contradictory information. The automation of advanced decision aids will bring the most significant contribution to the goal of producing timely intelligence. Aids will share (or be coupled with a logical copy of) the analyst's working space, but will never alter it without the analyst's approval.

To reach this ambitious objective, we defined two preliminary steps: the identification of an architecture allowing the synchronization of these aids between themselves and within the analyst's working space; and the identification of these aids.

Identification of an Architecture

One of the possible architecture is the use of the blackboard technology where different knowledge sources share a global working space (Ref. 11). Decision aids could then be encapsulated in knowledge sources. An investigation of the applicability of the blackboard architecture adapted to this problem was carried out and reported at Reference 12. The study revealed that a distributed blackboard architecture, as opposed to a centralized one, will better serve this problem by offering the possibility to split decision aids amongst several processors working in parallel. Although, this approach was feasible, the study recommended further investigation. We are then planning to investigate the applicability of a multi-agent distributed architecture (Refs. 13 and 14).

Identification of Advanced Decision Aids

We were in favor of an exploratory method to maximize our chance to come up with advanced decision aids that will be appreciated, useful, comprehensive while being computationally feasible. Consequently, several research activities have been initiated to enable us to acquire a better understanding and appraisal of the intelligence fusion task and its difficulties. The other advantage of the exploratory method is to verify the ability of some AI techniques in solving specific related problems (Refs. 15 to 19). Research activities are presently carried out by a team composed of the authors and other members of the research group. They include:

- *Recognition of Doctrinal Deployment Templates:* To automatically and interactively assist the analyst in aggregating enemy elements into higher level formations by matching templates with the known situation, and to suggest possible existence of unsighted enemy elements from those formations.
- *Combination of Redundant Information:* To automatically recognize and suggest potential sightings for merging or tracking.
- *Recognition and Identification of Enemy Activities:* To interactively assist the analyst in identifying major enemy activities based upon critical indicators which form negative or positive evidence of possible activities.
- *Representation and Justification of Analyst Expectations:* To allow for representation of analyst's expectations about unsighted enemy elements, and for automatic alerting if matching sightings occur (Ref. 20).
- *Terrain Visualization, Analysis and Modeling in a Virtual Battlefield:* To provide the necessary means to better assimilate, understand and analyze terrain topography and associated features in the context of a virtual battlefield situation (Ref. 21).

Conclusion

This paper has discussed an approach towards the development of decision aids tailored to the individual work of analysts in performing intelligence fusion. An incremental prototyping approach was adopted; it consists of two stages: (1) to develop basis decision aids by allowing analysts to use a set of predefined fusion operations, making them more productive while staying in full control over the analysis process; and (2) to gradually incorporate more *intelligent* decision aids to assist analysts by really contributing to the overall decision-making process.

This approach proved to be very appropriate. The first stage of the approach was successfully carried out. All basis decision aids were embedded in a powerful man-machine interface along with a graphical reasoning support system that allows analysts to visualize and modify their chain of reasoning. The man-machine interface has been included in a larger computing environment that supports cooperative assessment of the enemy situation from several intelligence workstations.

Basis decision aids rely on concepts that were fully approved by intelligence experts upon an abbreviated command post exercise simulating a three-hour scenario of high-level intensity conflict. Fusion operations were found relevant and useful to help achieve the resolution of the enemy *puzzle*. Most of the intelligence community who attended the exercise encouraged us to proceed to the *delivery* phase of the computing environment and to pursue further research on advanced decision aids. Towards this last end, we have highlighted in this paper two issues: one concerning the problem of selecting (or adopting) an appropriate architecture to allow the synchronization of decision aids and; the other one, concerning the exploration of specific research areas related to some aspects of the global intelligence fusion problem.

Acknowledgment

The authors are embedded to Mr. M. Parenteau and Mr. M. Cantin who contributed significantly to the

success of the first stage and to Mr. S. Lapointe, Mr. S. Woods and Mr. G. Vézina who participated in the definition of the research activities related to advanced decision aids.

References

1. Gauvin, M., Des Groseillers, L. and Roy, D., *Tactical Information Fusion Prototype: Research Project Definition*, DREV M-3068/91, Quebec, May 1991.
2. Centre de recherche informatique de Montréal,*Tactical Information Fusion Prototype Decision-Aid Subsystem Development Approach and Techniques*, Contract No. W7701-9-5219/01-XSK, Quebec, November 1990.
3. Headquarters Department of the Army, *FM 34-3 Intelligence Analysis*, March 1990.
4. Army Doctrine and Tactics Board, *Combat Intelligence*, B-GL-315-002/FT001, July 1988.
5. Headquarters Department of the Army, *FM-34-130 Intelligence Preparation of the Battlefield*, May 1989.
6. Edwards, P.H., *Data Fusion in the Land Battle Environment*, AGARD-CP-440, 1988.
7. Woolsey, R. B., "Application of Expert System Technology to Information Fusion System", *Application of Artificial Intelligence to Command & Control Systems*, Ed. C.J. Harris, Exeter, England, Peter Peregarinus Ltd., pp. 185-211, 1988.
8. Naylor, R. T., "Battlefield Data Fusion", *Application of Artificial Intelligence to Command & Control Systems*, Ed. C.J. Harris, Exeter, England, Peter Peregarinus Ltd., pp. 212-233, 1988.
9. Des Groseilliers, L., *Tactical Information Fusion Prototype: Basic Intelligence*, DREV M-3093/92, Quebec, February 1992.
10. Roy, D., Gauvin, M. and Parenteau, M., *Tactical Information Fusion Prototype: Manoeuvre Element Analyst Environment*, DREV M-3102/92, Quebec, November 1992.
11. Nii, H.P., "Blackboard Systems: The Blackboard Model of Problem Solving and the Evolution of Blackboard Architectures", *The AI Magazine*, pp. 38-53, Summer 1986.
12. DMR Group Inc., *Knowledge Representation for Tactical Information Fusion*, Contract No. W7701-1-0659/01-XSK, Quebec, May 1992.
13. Ferber, J. et Ghallab, M., "Problématique des univers multi-agents intelligents", *Actes des journées nationales du PRC-IA*, Toulouse, mars 1988.
14. Ghédira, K. et Verfaillie, G., "Approche multi-agent d'un problème de satisfaction de contraintes: optimalité et réactivité", *Twelfth international Conference*, Avignon, juin 1992
15. Baker, J. E. et al., "A C^3 Workstation Utilizing Value-Based Message Scheduling", *Proceedings of the 1992 Symposium on Command and Control Research*, Monteley, California, June 1992.
16. Morgan Peter D, "The Application of a Model of Adaptive Decision Making to the Collection and Analysis of Domain Expertise", *Proceedings of the 1992 Symposium on Command and Control Research*, Monteley, California, June 1992.
17. Zuidgest, R. G., "Multi-sensor data fusion and the use of artificial intelligence", *Twelfth international Conference*, Avignon, June 1992.
18. Bares, M. M., "La problématique de R&D liée au C3I", *Journée Thématique INFORMATIQUE DE COMMANDEMENT*, Direction des Recherches, Études et Techniques, Ministère de la Défense, Paris, juin 1993.
19. Chaudron, M. L., "Une étude de représentation de la connaissance structurelle dans MATIS", *Journée Thématique INFORMATIQUE DE COMMANDEMENT*, Direction des Recherches, Études et Techniques, Ministère de la Défense, Paris, juin 1993.
20. Williams, G. and Woods, S., "Representing Expectations in Spatial Information Systems: A Case Study", *Proceedings of the 3rd International Conference on Large Spatial Databases*, Singapore, June 1993.
21. Vézina, G., "Geomatics and Visualization: the Virtual Battlefield", *DND Workshop on Applications of GIS to Canadian Forces Mission Planning and Decision Support*, Ottawa, June 1993.

903158

The Case-Based Planning System: Using Case-Based Reasoning and Genetic Algorithms to Plan Spacecraft Operations

Rudolf Jetzelsperger
Software Kinetics Limited
Stittsville, Ontario, Canada
e-mail: jetz@crc.sofkin.ca

Peter Adamovits
Canadian Space Agency
Ottawa, Ontario, Canada
e-mail: adamovitsp@sp_agency.ca

#149664

Abstract

Closed environment systems like spacecraft require rigorous control over the use of resources. Over allocation of resources can result in catastrophic results in the entire system ranging from the loss of life to the destruction of an expensive spacecraft. As every activity in the spacecraft involves interaction with resources, both renewable and non-renewable, the planning of these activities become of paramount importance to the well being of the spacecraft and crew. As the complexity of the operation and resource interactions increase the ability of human operators to effectively plan operations diminishes. The domain of automated planning has been an active area of research for many years. One approach to the planning problem that has seen increasing popularity recently is the use of case-based reasoning. Case-based planning uses previously generated plans to meet similar situations or adapt previous plans to meet new situations.

This paper describes the reasoning elements of the Case-Based Planning System: a planning system that incorporates case-based reasoning and genetic algorithms to plan operations in the highly constrained environment of spacecraft operations. A robust plan specification model has been developed which enables a user to graphically specify complex constraints and task relationships. Case-based reasoning is used to retrieve appropriate existing plans that meet a user specification of a desired goal. If the retrieved plans do not meet the specification, the system uses a genetic algorithm to adapt the plans until the specification is met. Together these two techniques enable the efficient planning of complex operations in a highly constrained environment.

Keywords: case-based reasoning, case-based planning, genetic algorithms, planning systems, hybrid systems

Introduction

In the planning and scheduling domain, as the number of tasks and, constraints increase, the effectiveness of the human operator diminishes. Considerable research has gone into the planning and scheduling of tasks under complex constraints. The incorporation of various Artificial Intelligence (AI) techniques, to assist in the solution of constraint satisfaction problems, is a common approach. Many planning systems take advantage of the strengths of individual techniques. Unfortunately, the use of a single technique to solve complex constraints usually reveals the limitations inherent in the methodology, thereby limiting the effectiveness of the overall planning system. One possible solution to such limitations is to employ several reasoning strategies which together attempt to overcome each other's inherit weaknesses.

This paper describes the current implementation of the Case-Based Planning System based on the design described in [1].

The Planning System Architecture

The intent of the planning system is to assist a human spacecraft operator or spacecraft operations planner in performing their duties in a constraint bound environment. It is intended that the system when extended will be capable of performing in an environment that to some extent is changing and dynamic in nature. The planning system reacts to changes in the domain of application by learning and adapting to the changes.

The planning system relies on a *case base* (library) of previously used plans that evolve over time as the system adapts to the changing domain environment. The user interacts with the system through the *user interface* by graphically defining the plan behaviour by providing a *specification* of the problem or tasks to be accomplished. Based on this specification, the *plan retrieval* component retrieves a set of plans (perhaps several hundred) that match this specification. Using the same user specification the *plan selection* component performs a more precise analysis selecting the plan that best meets the requirements defined in the specification. If none of the retrieved plans fulfill the requirements of the specification, the selection mechanism *adapts* the plan using a *genetic algorithm* approach until an acceptable plan is created. The success of the selected or adapted plan is then *evaluated* by the operator and the evaluation information is used in the case *library update* facility. Details of how these systems interrelate are presented in following sections and is illustrated in Fig. 1.

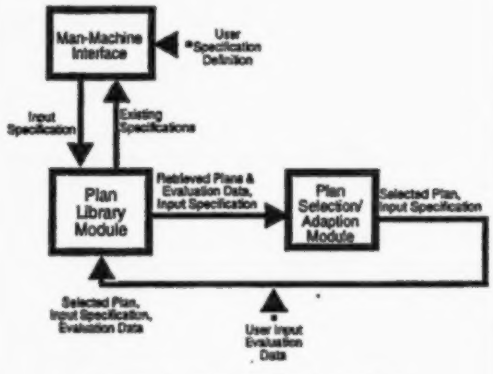

Man-Machine Interface
User *Specification Definition
Input Specification
Existing Specifications
Plan Library Module
Retrieved Plans & Evaluation Data, Input Specification
Plan Selection/ Adaption Module
Selected Plan, Input Specification
Selected Plan, Input Specification, Evaluation Data
User Input Evaluation Data

Figure 1 - System Architecture

Activity, Plan, Specification and Constraint Language

The planning system described in this paper includes several reasoning systems that rely on a consistent definition of the data types for their effective communication and operation. The major data types are described here to assist the reader in the explanations that follow. Several additional hidden data types are also defined however these have more implementation than conceptual significance.

An *activity* is an event that serves some useful purpose. In our system the activities are related to spacecraft operation management (activities such as replacing or repairing components of a spacecraft, doing regular maintenance etc.). The granularity of activities is operator defined or created. That is, the planning system does not assume a particular granularity of activities. They can be high level events such as a shuttle launch or as low level as pushing a button.

In our system an activities are defined by *specifications*. A *primitive activity*, one which is atomic, is called a task and is represented by a *primitive specification*. A *complex activity*, is an activity which is composed of other activities and is represented by a *complex specification*.

Primitive activities' atomic characteristics are defined by *tasks*. A *task* has a notion of the duration of time it requires to complete but not when or where it is to be used. Tasks have a knowledge of the resources they require but may or may not have associated with them specific amounts of the resource required (for example it may be a function of some parameter external to the task). For further clarification on the terms used the reader is referred to Georgeoff [2].

In general terms a *specification* is a description of the requirements of an activity. Associated with a specification is a list of *constraints* which provide the structure to the activities that the specification is composed of. There are two types of specifications currently used by the system: *primitive specifications* and *complex specifications*. *Primitive specifications* are the specifications associated with a specific task (e.g. its resource requirements, preconditions, etc.).

Constraints are definitions of relationships between activities (inter-activity constraints), and limitations on a single activity. Some of the inter-activity constraints currently defined are: (a before b) (a immediatelyBefore b) (a after b) (a immediatelyAfter b) (a overlaps b) (a precedes b) etc. Inner-activity constraints are subdivided into two categories: resource utilization constraints (e.g. electrical power usage) and temporal constraints (e.g. activity a must be complete by some time t).

Associated with all constraints are weights which determines the relative importance of the constraints. Not all constraints are considered to have the same priority. Some constraints are soft in nature while others are hard and must never be violated. For example it may be preferable to exit eclipse with not less than 1.2 volts in any battery cell however if most cells are at 1.3 volts and one cell is at 1.15 volts this may not be sufficient grounds to abort the plan based on the constraint violation. In another instance, it is never possible to violate the eclipse start point or duration to create a more optimal plan.

There is a clear distinction between a *plan* and a *specification* in this system. A *plan* is a structured sequence of *tasks* that have associated relative start times that are combined to perform some activity. Unlike a specification a plan has no defined structure. A plan is merely an instantiation of a specification, an ordering of tasks that meets the requirements of one or more specifications. The activity that a plan is designed to perform is determined by the specifications that the plan satisfies. For instance a plan may satisfy more than one specification and could therefor be used for more than one activity (e.g. repairing a power supply anytime and repairing a the power supply before a given point in time could be represented by the same plan). A plan always satisfies at least one specification.

Plan Specification

The user interacts with the system by defining the operations to be completed thereby creating a plan specification. The user graphically specifies a desired behavior in terms of previously defined specifications. The specification is composed graphically using a PERT chart syntax where the nodes represent previously defined specifications and the arcs represent inter-specification constraints. Inner-specification constraints are also graphically represented inside the nodes. In this fashion the operator can define a new activity or operation based on previously defined activities or operations.

The user is also provided with the capability to define new atomic activities. The operator merely defines the resource utilization and the activity duration. When the operator creates a new atomic activity the system automatically adds a dummy plan that fulfills the specification of the new activity. This ensures that every specification in the system has at least one representative plan.

Once the specification of a plan is complete the specification is processed by the plan retrieval component of the system to retrieve a plan that matches the specification.

Plan Retrieval

The plan retrieval component is responsible for the retrieval of a set of plans that match an input plan specification. The retrieved plans can match the input specifications wholly or partly. The plan retrieval component employs case-based reasoning [3 and 4] to retrieve plans that match an input specification. Plans are stored in the case-base and are indexed by the most specific specifications that they fulfill. The case-base indexing hierarchy is based on subset/superset relations. The case-base indexes are organized such that the most general specifications are at the top level of the index hierarchy and the most specific at the bottom. This implies that the top level of indexes will consist of primitive specifications which represent atomic activities and cannot be further decomposed. The indexing structure is illustrated in Fig. 2. The indexing mechanism provides for quick but robust retrieval of plans which match the input plan specification

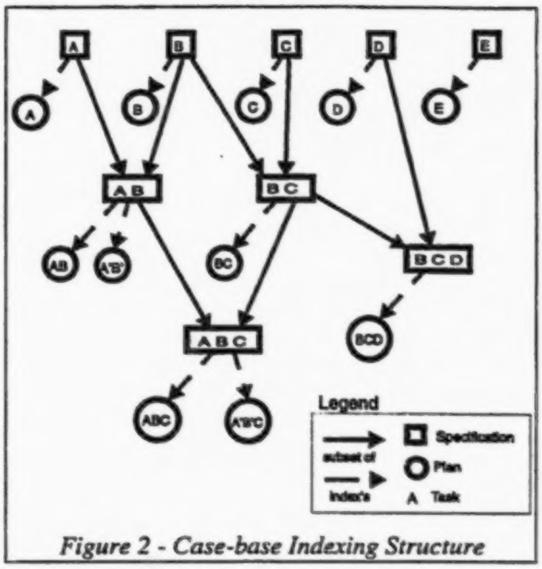


Figure 2 - Case-base Indexing Structure

Each index is composed of a specification. Associated with the specification are the specification's children and a list of the plans that the specification indexes. These plans fulfill the specification but none of the specification's children. That is, the current specification is the most detailed specification that the plan fulfills. The specification's children represent more specific specifications than the parent specification. A specification can index more than one plan but every plan is indexed by at least one specification.

The plan retrieval algorithm is highly parallel in its execution. That is, the retrieval of matching plans can be carried out independently in several directions simultaneously. The search starts by determining the primitive specifications that the input specification is composed of, and proceeds in parallel down the index graph for each primitive specification. Each primitive specification search can be considered an independent search thread. Each index specification that the input specification matches the index specification will add its associated plans to the retrieved plan list and test the index specification's children for matches. A search thread will terminate when an index specification does not match the input specification. When an index specification fails the match (e.g. the input specification is a superset of the input specification) the index specification will add its associated plans to the retrieved plan list but will not propagate the matching to its children. When the retrieval algorithm is completed the retrieved plans will include plans that satisfy subsets of the specification as well as supersets.

The aim of the plan retrieval process is to find the most specific specifications in the indexing scheme that match the input specification. Once all the relevant plans have been retrieved they are processed to ensure that they *cover* the input specification. To *cover* the plan specification completely each retrieved plan must contain a representative of every task specified in the input specification and contain no extraneous tasks. If the plan does not contain a task that is in the input specification the task is added arbitrarily to the plan. If the plan contains extra tasks that are not in the specification, they are removed from the plan. Since the input plan specification is created using existing specifications, it is impossible to specify, in the input specification, something that is not in the case-base indexing mechanism.

Each case in the case-base contains a plan, and its associated evaluation data. Currently the evaluation data contains only the times used and times successful counts. This data gives some indication of the relative quality of the plan.

The input specification, retrieved plans and their associated evaluation data are then sent to the plan selection and adaptation module to determine or create a single plan that fulfills the input specification.

Plan Selection and Adaptation

The plan selection and adaptation module accepts as input a collection of plans and a specification. The output of the Plan Selection Module is a plan or plans that fulfill(s) the criteria specified in the input specification. A plan fulfills the input specification if it meets all the constraints in the input specification. The plan selection and adaptation module assumes that the input plan specification is a valid specification and that there is a possible solution. If this assumption is invalid, the selection process will determine the best possible fit for the specification. This result, however, will not necessarily meet all the criteria in the specification.

The plan selection and adaptation module evaluates each plan in the input plan collection with respect to the input specification. The evaluation of a plan with respect to a specification involves the unification of the tasks in the plan with the tasks in the specification and the evaluation of the constraints as they relate to the tasks in the plan. Each of these constraints has an associated weight which determines the relative importance of the constraint. These constraints are evaluated and a score is assigned to the plan with respect to the fulfillment of the specification (range {0...1}). If the plan meets all the constraints in the specification then it is assigned a value of 1 and is said to fulfill the specification. If there is a a plan that fulfills the specification the selection process is terminated and the fulfilling plan is returned.

The more likely case is that there are no plans that fulfill the input specification. When this happens, the system will attempt to adapt the input plans until a plan that fulfills the input specification is created. The adaptation process uses a genetic algorithm to adapt the plans. The genetic algorithm used is based loosely on Goldberg description of the genetic algorithm [5]. Fig. 3 provides a general overview of the genetic algorithm used by the plan selection and adaptation module. Table I maps the natural terminology as defined by Goldberg [6] into the CBPS planning architecture terminology:

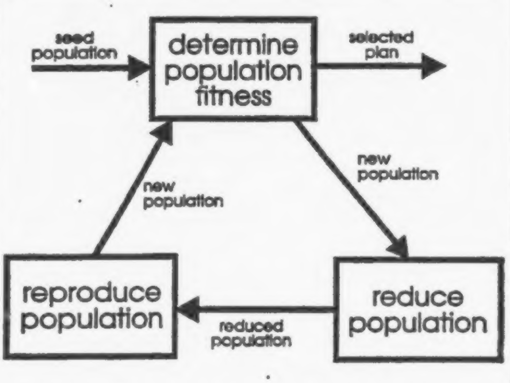


Figure 3 - Genetic Algorithm

Plans that do not fulfill the input specification are given a fitness value that is proportional to the plans *fitness distance* from a fulfilling plan. This distance is measured by each constraint in the specification and is cumulative across the entire specification.

The input plans are used as the seed population instead of randomly determining the seed population as described by Goldberg. This enables the starting population to build from established building blocks [7 to 9]. The fitness of a plan is determined by the evaluation of the input specification. The evaluation criteria is based on weighted constraints evaluation formula. The formula returns the how well the plan fulfills the specification (range of 0-1). Currently the fitness function for the plan selection mechanism is a function of the number of constraints that are satisfied by the plan. The more plan specification constraints that are satisfied the better fit the plan is to the plan specification. Each constraint has a weight associated to it. The weight of the constraint determines the importance of the constraint. Soft constraints (such as optimization constraints) have very little weight while hard constraints (such as resource utilization constraints) have high weights. Each constraint will indicate the degree of satisfaction (range 0-1). This enables the system to more accurately differentiate between similar solutions to determine the best solution.

Table I: Terminology Mapping

Natural Term	Planning Domain Term
population	a collection of plans
being	a plan
chromosome	a plan
gene	a task in a plan
entity	a plan
allele	a task's attributes
locus	the position of the task in the plan
genotype	a collection of plans

Unlike the genetic algorithm defined by Goldberg, our GA employs a steady state population model. That is, there is a population threshold that restricts the size of the population at each generation. At each generation the least fit plans are killed off. This ensures a reasonable population size. To provide variety in the population the algorithm removes duplicate entities from each population generation.

Another deviation from Goldberg's algorithm is the propagation of parents. In the Goldberg algorithm the parents are killed off every generation and the only representation of the parents are their children. This can cause the loss of more fit entities i.e. when the parents are more fit then their children. In our algorithm, parents remain in the population until their fitness eliminates them. Because of the steady state population, at each generation only the most fit plans are kept and the remaining population are killed off. This ensures that the overall fitness of the population will never decrease. To prevent endless cycling and ensure a reasonable response time, a generation threshold is used to limit the number of generations that are produced. Once this threshold has been reached the plan adaptation module returns the best plan of the last generation of plans.

In our system the mutation operation is much more important than in Goldbergs. This is due to the fact that the relative positions of the tasks in the plan might not be present at all in the starting population. This implies that with crossover alone there is no chance of a solution to some specifications. For example: if there was a constraint that stated a task must start at time t, but none of the input plans had tasks that started at time t, there would be no way with crossover alone that a solution would be reached.

The population growth in each generation increases significantly to provide variety in the population. Given a population of size ***n***, in the next generation the population will grow to ***5n*** through propagation by mutation and crossover (***2n*** for mutation and ***2n*** for crossover for a total of ***5n***). Of these ***5n*** only the most fit ***n*** entities will be carried over to the next generation. The crossover operation is based on Davis' Partially Matched Crossover (PMX) [10 and 11]. For a pair of arbitrary plans, similar tasks are lined up, a random crossover point is determined, and the task/gene attributes are exchanged. Fig. 4 describes the crossover operation. There are two forms of mutation that are used in the algorithm: plan and task mutation. Plan mutation

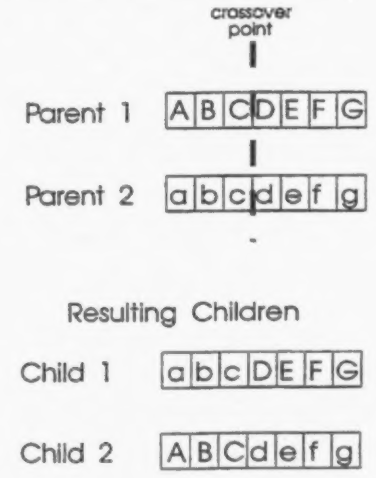
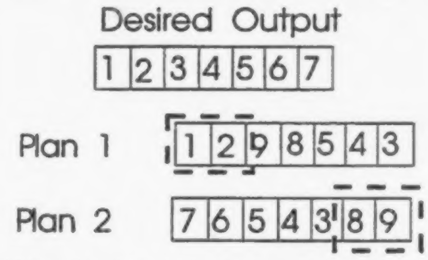


Figure 4 - Crossover Operation

selects two random tasks in the plan and exchanges their attributes. Task mutation modifies the task attributes (namely start time).

One of the most important conditions for convergence of the genetic algorithm is the accurate weighing and evaluation of the constraints in the input specification. The weight associated with the constraints has a direct impact on the order of constraint satisfaction. For instance, in a specification that requires the serial ordering of n tasks with no slack time between the tasks, the constraints on the tasks at the beginning of the task list are more important (carry a larger weight) than the constraints on the tasks at the end of the task list. The reasons for this have to do with the distance measure from the fulfillment of the input specification. For example: a plan with the last two tasks adjacent is further from fulfillment than a plan with the first two tasks adjacent. Fig. 5 describes this example.



Although plan 1 and plan 2 satisfy the same number of constraints, plan 1 is closer to the desired output than plan 2 because in order to achieve the desired output the satisfied constraints in plan 2 must be violated

Figure 5 - Fitness Distance

The cycle of reduction and production of the input population continues until either a successful plan is created or the generation threshold has been reached. In either case a plan or set of plans are returned and passed to the instantiation module.

Case Library Maintenance

The case library maintenance component is responsible for maintaining the indexing scheme in the light of new plans and specifications being added to the case-base. When a new specification/plan pair is added to the case-base a new index specification is created with the associated plan and specification. This new indexing specification is then inserted into the indexing scheme. The insertion algorithm starts with the most general specifications (primitive specifications). At each level in the indexing scheme the new index specification is compared against the existing index specifications. If the new specification is equal to an existing specification, the associated new plan is added to the existing index specification and the algorithm terminates. If the new specification is a superset of the existing specification, the new specification is compared against all the children of the existing specification. If the new index specification is a subset of any of the specifications, the new index specification is inserted between the parent and the subset children. This is achieved by adding the new index specification to the parent as a child and removing the subset children from the parent and adding them to the new index specification. Fig. 6 diagrams an example of this process. If there are no children, the new index specification is added as a child to the parent specification. Fig. 7 describes an example of this case.

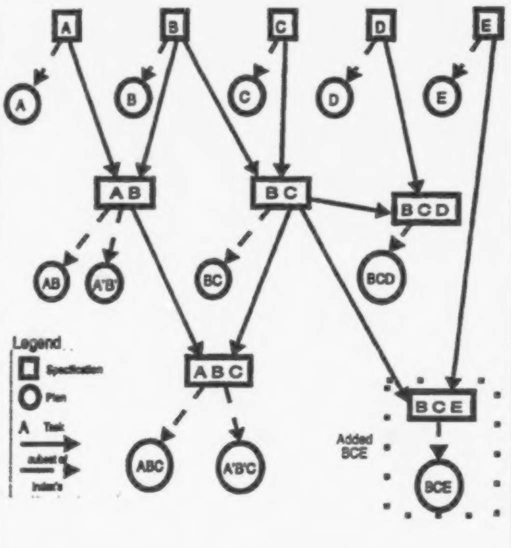


Figure 6 - Index Addition Example

Status of Present Development

The system described here represents the fourth generation of a project investigating the application of AI techniques for resource management and task planning/scheduling. Previous versions of the system relied primarily on case-based reasoning approaches and a case library to record and store the relevant past histories of successfully applied plans [12 and 13]. In terms of the plan and schedule definitions used in this paper, these previous versions also maintained a library of *plans* instantiating and verifying their appropriateness on a model of the system prior to their application in the target application domain [1]. There were several failings of varying degrees with these previous attempts. However, of particular interest was the ability of the system to adapt a known plan to alternate applications based on the new user specifications and to address the needs of large case libraries (e.g. 10,000 or more case elements). The specification language and the addition of Genetic algorithms are considered significant improvements to the retrieval and adaptation and selection aspects of the system respectively.

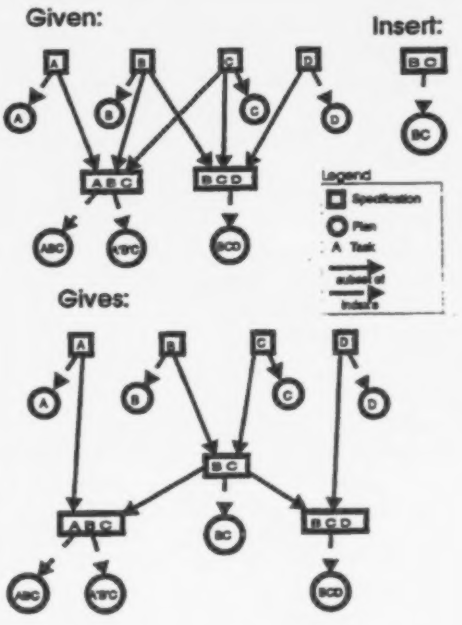


Figure 7 - Index Insertion Example

The genetic algorithms approach *learns* and *adapts* to the needs of the system as the case library changes and evolves. Additionally, the control mechanisms for the genetic algorithm element in the system is domain independent and does not require an extensive amount of hand tailoring and adaptation to suit differing application domains. This approach is a key component in the implementation of the planning/scheduling system.

The current version of the planning/scheduling system was developed in Digitalk's Smalltalk/V for Windows. A multiple developer/software management system from Object Technology International (OTI) was also used to provide version control and librarian facilities missing in the base Smalltalk environment. Finally, under a separate project, OTI is developing a Smalltalk class library of rule base and case base software primitives to assist in the development of rule based and case based software systems. Although these later systems are only now in beta test, they will be forming the basis of the latest version of the planning/scheduling system. The constraint based specification language used in this work developed for earlier versions of the planning/scheduling system has been further refined in the present system.

Summary

We have described an advanced planning system that has recently been developed within the Canadian Space Agency. The system includes several reasoning elements, the combination of which is believed to be unique to this system.

References

1. Adamovits, Peter; Jetzelsperger, Rudi. 1993. *Using Coordinated Reasoning Strategies for Constraint Satisfaction in Spacecraft Operation Planning and Scheduling*, Proceedings of the 4th Workshop on Artificial Intelligence and Knowledge Based Systems for Space, Noordwijk, The Netherlands

2. Georgeoff, Michael P., 1987, Planning, *Annual Review of Computer Science*, Volume 2, Annual Reviews Inc., Palo Alto Ca.

3. Schank, R. C., 1982, *Dynamic Memory: A Theory of Learning in Computers and People*, Cambridge University Press, Mass.

4. Hammond, K. J., 1989, *Case-Based Planning: Viewing Planning as a Memory Task*, Academic Press, San Diego, Ca.

5. Goldberg, David, E. 1989, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley Pub., Mass, pp. 10-14

6. Goldberg, David, E. 1989, *Genetic Algorithms in Search, Optimization, and Machine Learning*,

Addison-Wesley Pub., Mass, pp. 21-22

7. Bethke, A, D. 1981, *Genetic Algorithms as Function Optimizers*, (Doctoral Dissertation, University of Michigan) Dissertation Abstracts International.
8. Holland, J, H. 1987, Genetic Algorithms and Classifier Systems: Foundations and Future directions. *Genetic Algorithms and Their Applications: Proceedings of the Second International Conference on Genetic Algorithms*, pp. 82-89.
9. Goldberg, David, E. 1989, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley Pub., Mass, pp. 41-45
10. Davis, L. 1985, Job Shop Scheduling with Genetic Algorithms, *Proceedings of an International Conference on Genetic Algorithms and Their Applications*, pp. 136-140
11. Goldberg, David, E. 1989, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley Pub., Mass, pp. 170-172
12. Oppacher, Franz; Deugo, Dwight. 1988. *A Dynamic Case-Based Planning System for Space Station Application*, 4th Conference on Artificial Intelligence for Space Applications, NASA Conference Publication 3013, Huntsville, Alabama
13. Adamovits, Peter; Ulug, Selim. 1991. *A Case-Based Planning System for Spacecraft Electrical Power System Control*, IAF, 42nd Congress of the International Astronautical Federation, Montreal, Canada, IAF 91-226

906227

DND Knowledge-Based Systems and Robotics Workshop, Nov. 1993

Virtual Telerobotic Control

Anu Rastogi[1], Paul Milgram[2], David Drascic[3]
Ergonomics in Teleoperation and Control (*Etc*) Lab
University of Toronto, 4 Taddle Creek Road, Toronto, ON M5S 1A4

Julius J. Grodski[4]
Defense and Civil Institute of Environmental Medicine
North York, ON M3M 3B9

#149666

Abstract

For teleoperation, the operator effectiveness can be improved by providing graphical aids, but these aids require a pre-built model of a remote site, which may not be available for unstructured environments. This paper presents an interface called ARGOS (Augmented Reality through Graphic Overlays on Stereovideo) for teleoperation to improve understanding of spatial relationships at the remote site by the operator, probe the remote site for spatial information, and provide means for interactively modeling the remote site for graphical support. In Virtual Control described here, the operator controls the robot through graphical simulation, and uses ARGOS to deal with teleoperation in unstructured environments.

Introduction

The demand for robotic operations in remote unstructured environments is growing, in applications such as battlefield robotics, (nuclear) waste cleanup, construction in space, etc. Such tasks are not foreseen to be achievable with high degrees of autonomy, due to cost constraints, technological unfeasibility, and/or high levels of operational uncertainty, and will consequently require mediation by a human operator (HO). In general, the role of the HO in teleoperator systems is expected to range anywhere from continuous closed loop manual control to monitoring semi-automated systems with some degree of local control autonomy. The latter extreme, which in essence characterises the nascent field of "telerobotics" [Sheridan, 1992], can be distinguished from highly automated industrial robotics by the larger uncertainties in the operating environment, which make it difficult to develop hard coded operational software.

Teleoperation in Unstructured Environments

One way to characterise potential teleoperation scenarios is according to the degree of prior knowledge about the work environment. At one extreme, the robotic environment and the objects being manipulated are largely invariant and well known in advance, and the operational procedures are prespecified. In this case preprogrammed operations are feasible and human supervision is required only for abnormalities, which may occur within the system or at the remote site. The other extreme refers to operations where the environment is not fixed and, even though the functions of the teleoperator may remain the same, the constraints can change and uncertainties are much larger. In such systems human supervision is required, at a level of involvement which depends upon the autonomy of the system with respect to the task. Even though the operator may be required to perform at a higher level of control, good visual, auditory, and perhaps even haptic, cues may nevertheless be required so that the operator will have a sense of the remote environment being veridically represented.

Any system in which the HO is assisted by some level of automation therefore requires various sensors, which contribute to the creation and maintenance of a model of the environment, which in turn is necessary for machine support [Schebor & Turney, 1991]. Even though such systems are capable of assisting the operator substantially, the cost is potentially very high and they may be robust for only small variations in the environment within which the machine intelligence is capable of working. Teleoperation systems based on machine vision are currently capable of recognition of only predefined objects in the scene [Spofford et al, 1991] and thus fail to provide the flexibility needed for teleoperation in unstructured environments.

To overcome such problems with highly autonomous systems, realistic teleoperation tasks should be suitably allocated between the human operator and machine intelligence [Burtnyk & Greenspan, 1991; Zhai & Milgram, 1992], so the respective capabilities of each are appropriately utilised. The essence of such a division of responsibilities is that humans are more suited for higher level perception and reasoning, task conceptualisation, understanding the environment and dealing with unusual circumstances, while machines are good at low level sensing and control , precision, reliability and computationally intensive tasks.

---

[1] Email: anu@argos.rose.utoronto.ca
[2] Email: paulm@argos.rose.utoronto.ca
[3] Email: drascic@ie.utoronto.ca
[4] Email: jul@dciem.dnd.ca

Our approach to telerobotic control seeks to allocate tasks such that the HO can perform optimally and in harmony with the machine's capabilities. It helps the human to generate and update remote environmental models on-line, monitor interactions of the robot with its environment and control the telerobot in an ecologically sound manner. The intended result is a flexible and robust teleoperation system from the viewpoint of environmental uncertainties and providing shared control between the operator and the automated support system.

Telerobotics by ARGOS

Telerobotic control by ARGOS (*Augmented Reality through Graphical Overlays on Stereovideo*) (Milgram et al, 1993) has two objectives: 1) To develop an optimal interface for control by the human operator, and 2) To provide means for the operator to interactively direct the modeling of the environment for on-line machine support. In achieving the latter, the operator is allocated the function of task planning and conceptualisation, performing higher level control of the telerobot, and communicating relevant environmental constraints to the machine. Upon obtaining an adequate world model, under direction of the human operator, the machine performs the lower level control of the remote manipulator, to support the higher level control of the operator. This is accomplished using ARGOS, together with a graphical simulation (virtual) of the telerobot, which will act as a display and control mediator for teleoperation.

ARGOS is a display and remote probing interface using calibrated stereoscopic graphical overlays on a remote stereoscopic video view [Milgram et al, 1991]. The remote scene contains a calibration object of known dimensions, which is used for calibrating the remote scene and making the remote stereovideo view amenable for "probing". Stereographical tools are used to determine remote world coordinates for points or objects in the scene, which can then be used for limited modeling of the remote environment. Once a model of an object in the remote video scene is created, the computer is able to enhance the outlines and further can support the operator with, for example graphical simulation of planned operations.

The major components of ARGOS technology are 1) Stereoscopic Video and 2) Calibrated Stereographics. These are discussed below, in terms of their applicability for teleoperation.

1) Stereoscopic Video

Most teleoperation systems use video cameras to provide feedback about the remote work-site. Although monoscopic video provides many important depth cues, such as linear perspective, interposition, relative size and shading, for tasks requiring finer spatial judgment these are often insufficient, and lack of binocular depth cues (i.e. stereopsis) can impede the operator's task performance. Although the HO does acquire knowledge about depth relationships at the remote site through learning [Kim et al, 1987], this typically occurs only for repetitive telemanipulative tasks [Drascic 1991]. To circumvent this problem, stereovideo has successfully demonstrated superior performance by the operator in many cases [e.g. Drascic & Grodski, 1993; Pepper et al, 1981]. Prior use of stereo video has also been found to reduce operator training times in using monoscopic cues [Drascic, 1991], and is much better in detecting slopes and gradients in the environment and reducing visual noise [Merritt, 1988]. Though retinal disparity cues are often considered advantageous only for complex and nonrepetitive tasks, these are precisely the kinds of task which characterise the operational environments being addressed by our research.

2) Calibrated Stereographic Overlays

There are three purposes for overlaying graphics in this application : 1) as a tool for "probing" the real remote environment visible on video, 2) for enhancing video images through real object overlays, compensating for degradation due to occlusion of objects, poor video quality and bad lighting conditions, and 3) for overlaying realistic looking but non-existent graphic objects, so that they appear to be a part of the video scene. The latter two applications are possible only for objects with previously defined models or those acquired on-line through mediation by the HO. In all cases, the real-world scene is enhanced by the virtual graphic overlays, creating an *augmented reality*.

Most display systems which use graphical overlays on monoscopic video calibrate graphics to remote world coordinates, placing the graphical objects in the same perspective view and location on the monitor as the real object in the remote video scene. Such monoscopic calibration establishes a one-to-one coordinate space mapping from the graphical world coordinates to the remote world coordinates, that is, any point specified in the virtual monoscopic graphical world has a unique identifiable point in the remote scene, which can be graphically depicted on top of the monoscopic video scene (since its perspective view is also known). With this capability, the virtual (graphical) manipulations of robot which use point specification are specified to the remote robot [Browse & Little, 1991]. However the reverse for monoscopic calibration is not true, as there is one-to-many coordinate space mapping of points from the real space as visible on monoscopic video to graphical space, causing ambiguity. This precludes any possibility of probing or conveying information from the video scene of the real world to the virtual (graphical) space. Consequently, only qualitative judgments can be made based on the monoscopic video view.

Using a stereoscopic video view of the remote site with ***stereoscopic*** graphics, ARGOS establishes a unique bidirectional one-to-one mapping of coordinate spaces between the virtual world and the remote world viewed through stereovideo (Fig 1). As a result, the spatial information after calibration, can be quantitatively transferred from the stereovideo scene to the graphical world and hence from real to virtual space.

This concept has important implications for interactive modeling. A coordinate point in the stereovideo view can be unambiguously defined, by moving an overlaid stereographic (virtual) pointer (discussed later) in stereovideo space on the monitor, to match the screen coordinates and retinal disparity of the video object or point. The operator only needs to discriminate between the relative depth of the pointer and the video object [Drascic & Milgram, 1991]. In comparison with a monoscopic system, the ARGOS virtual pointer, when calibrated to stereo cameras, uses extra information contained within the binocular disparity to compute the actual depth information of the virtual pointer.

This technology can also be applied to separate (typically orthogonal) two camera viewpoint systems, as proposed for example by Cannon & Leifer [1991] or Oxenburg & Landell [1988]. Among the disadvantages of such systems, however, are that the operator must integrate information from the two windows and that there should be two controls for the two views to maintain display control compatibility, with the operator repeatedly switching between these. This forces the operator's cognitive behaviour in multiple views to a knowledge based level, in comparison with skill based behaviour for single viewpoint stereoscopic cameras. In addition, multiple views do not offer a coordinated and integrated control in 3D space, in contrast to stereo viewing.

Description of ARGOS

ARGOS is based on the superimposition of stereographics onto a stereovideo display of the remote world. The remote scene and cameras are calibrated by means of a calibration object. In our system the robot serves as the calibration object, since it is present at the remote scene and its geometric parameters are well known. After stereo calibration, spatial location of any point in the remote space can be found relative to the coordinate system of the remote world by simply pointing at it with a graphic cursor. Since this is done in stereo, these points can be specified by the operator in the real video scene directly and unambiguously.

By specifying multiple points in the real world 3D space, virtual objects such as cubes, planes etc. can be created easily, with their corresponding position and size in the real world known immediately. These overlaid images can then be interactively moved about to a desired location within the video image, to give the impression that they are actually at the remote site. Whenever the camera position changes relative to the remote world coordinate system and the displacement is not known, the calibration is done again on-line, to maintain robotic accuracy.

Interactive Modeling with ARGOS

In robotic systems supported by graphical simulation, a complete model of all robot elements, as well as the remote environment, is required before the

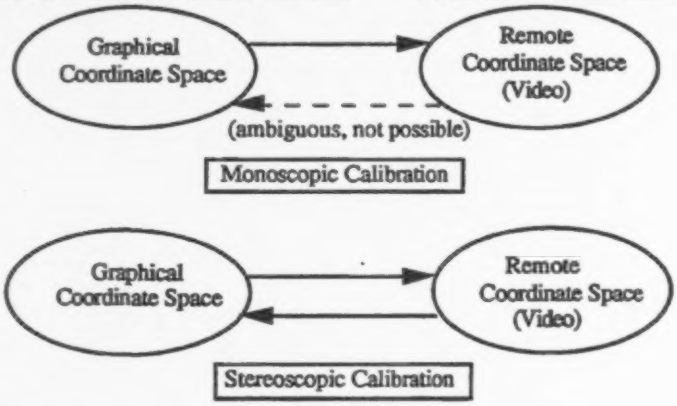

Graphical Coordinate Space
Remote Coordinate Space (Video)
(ambiguous, not possible)
Monoscopic Calibration
Graphical Coordinate Space
Remote Coordinate Space (Video)
Stereoscopic Calibration

Figure 1 : Comparison of monoscopic and stereoscopic calibration

actual operation. Such remote world models can not be prespecified with accuracy, if the environment is varying and unstructured, thus limiting the scope of operation in such environments. To solve this problem,the concept of *interactive modeling* is proposed as a means to enabling the human operator to convey limited amounts of data about the video scene to the computer, to gradually develop and refine a quantitative model of portions of the remote world, but not for deriving an accurate model of the entire remote scene. The interactive modeling tools presented in the following section permit the operator to specify necessary spatial information for defining teleoperation boundaries and obstacles, as well as waypoints for lower level semi-automated control of the telerobot by the machine.

ARGOS Tools

The tools provided by ARGOS can be classified into "probing tools" and "enhancement tools". All of these tools have been designed both for improving the HO's comprehension of the remote environment and for *interactive modeling* of the remote world.

The most basic probing tool is the *virtual pointer*. This is a stereographical cursor that can be positioned anywhere in the stereovideo scene, the location of which, can be specified in video space through the stereo calibration procedure outlined above [Drascic & Milgram, 1991].

The *virtual tape measure* is used for measuring distances between points in the remote stereovideo scene. It is generated by clicking a start point with the virtual pointer and dragging a virtual line of calibrated length through the video image to a designated end point. Related to this are *virtual landmarks*, which are graphical objects of known length, or known separations, superimposed on the video scene to enhance the HO's ability to judge absolute distances. Research has shown that such simple graphic aids can be very effective for this purpose [Milgram & Krüger, 1992].

Virtual planes are generated by specifying three or more coplanar points with the virtual pointer. One important application of these planes is for restricting movement of simulated objects. For example, a graphical model of a (virtual) robot interactively placed on a real table would not ordinarily have any knowledge about the planar constraints of the tabletop. A straightforward way of conveying such information to the computer would be through interactive modeling (as described above) by means of virtual planes.

Virtual objects, which are premodeled according to particular geometric specifications, can be superimposed on stereovideo at designated locations and at specified orientations and appear as if they are really present at the remote site. Related to these are virtual *encapsulators*, which are wireframe shapes created on the remote stereovideo scene to encapsulate, or highlight the edges of, real objects. The same modeling, location and orientation data as for virtual objects are needed for these

Virtual trajectories are prespecified graphical trajectories of prescribed robot motions, added to the image of the real robot at a particular initial configuration, and specify the real trajectory for the robot to follow. These can be used, for example, for path planning purposes, by placing them in video space and verifying them for their accuracy in relation to the actual (unmodeled) worksite. As an example, the trajectory for a particular picking and place task can be added and displayed, for verifying its accuracy.

A subclass of the virtual trajectory is the *virtual tether*, which was developed as a perceptual aid for manual telemanipulation tasks. A virtual line, or "tether", is drawn between the end effector and its intended target. As the manipulator moves, it remains "tied" to the target through the tether. Research has shown that use of such a tether was able to improve accuracy in an experimental peg-in-hole task, relative to the case of stereovideo alone with no tether [Ruffo & Milgram, 1992].

Finally, a full 3D stereographic model of the remotely controlled robot, or a *virtual manipulator*, has been developed. By superimposing a model of the robot at the remote site onto the real robot, the graphical model can be manipulated within the real (complex, unstructured, unmodeled) 3D work space. This concept is discussed in more detail below.

Graphic Overlays for Robot Simulation

Task related information by means of graphic overlays has been used by others for providing a variety of data from the remote site, such as contact forces, proximity information, orientation of gripper etc. [e.g. Ince et al, 91; Kim & Bejczy, 1991]. Graphical displays can also be used for presenting complex information about the state of the teleoperator system and environmental constraints. However, it is important to note that, although additional relevant information can in principle be helpful, any new type of information has to be cognitively integrated into the display system as a whole. This may increase the training costs as well as increase screen clutter, which may in turn lead to bad judgment due to perceptual inaccuracies. On-screen graphic "aids" are helpful only when they are successfully integrated into the real video scene and able to highlight only pertinent task-related points in the scene, with minimal cognitive processing.

Graphical simulation of robot operations is a well-known technique in path planning [e.g. Brunner et al, 1993; Kim et al, 1993]. In particular, this allows specification and verification of the planned robot path, without the expense of actual trials or the hazard of potential collisions in real environment, before actual execution by the remote teleoperator. Conventionally, such simulations are carried out off-line, in a totally modeled environment, which of course also allows visualisation of planned operations from any desired viewpoint.

Our system allows replication of moves of the robot in the *same* visual environment as the real robot, most notably superimposed on the real remote video scene. (Using our limited interactive modeling capability, we are also capable of conducting some degree of simulation based on a completely modeled world; however, that topic is beyond the scope of this paper.) The advantages of this superimposed modeling approach are that the effort of acquiring a detailed world model is not necessary and the ability of the operator to visualise the simulated tasks is greatly enhanced by the fact that the visual background coincides with the actual remote worksite.

It is important to make a distinction between our system and other related efforts. One of these is the JPL "advanced graphics interface" for telerobotic servicing and inspection [Kim et al, 1993], which provides for analogous preview, prediction and on-line visualisation capabilities, but not with stereoscopic displays for both graphics and video. Another is the German ROTEX project [Brunner et al, 1993], which is quite sophisticated in simulation and world modeling capabilities, but presents the graphics and video on adjacent displays, rather than integrated within a single display. Cannon & Leiffer [1991] have also proposed a system for point-and-direct telerobotics; however, as pointed out above, their system is based on a two camera view system, rather than an integrated stereo display. Finally, Browse & Little [1991] have developed a similar virtual control prototype, but again using only monoscopic displays.

Virtual Control with ARGOS

Under virtual *control*, the remote robot is controlled by the operator by means of a virtual (graphical) robot [Zhai & Milgram, 1991]. The virtual robot can be used, for example, to specify a trajectory, for later emulation by the remote robot.

Using ARGOS as a display and control mediator, the telerobotic system is initialised by calibrating the stereo cameras. A stereoscopic graphic robot model is then overlaid on the remote stereovideo image of the actual robot on the monitor. The robot joint angles are read and written by the graphics workstation through serial port. Initially the virtual robot's joint angles are initialised from the robot at the remote site. Afterwards joint angle commands are sent from the workstation to the robot for execution. A 6 degree-of-freedom (d.o.f.) input device is used to control the end-effector of the virtual model of robot with the remote robot following the virtual robot either concurrently or belatedly.

The graphical model is drawn in wire-frame when

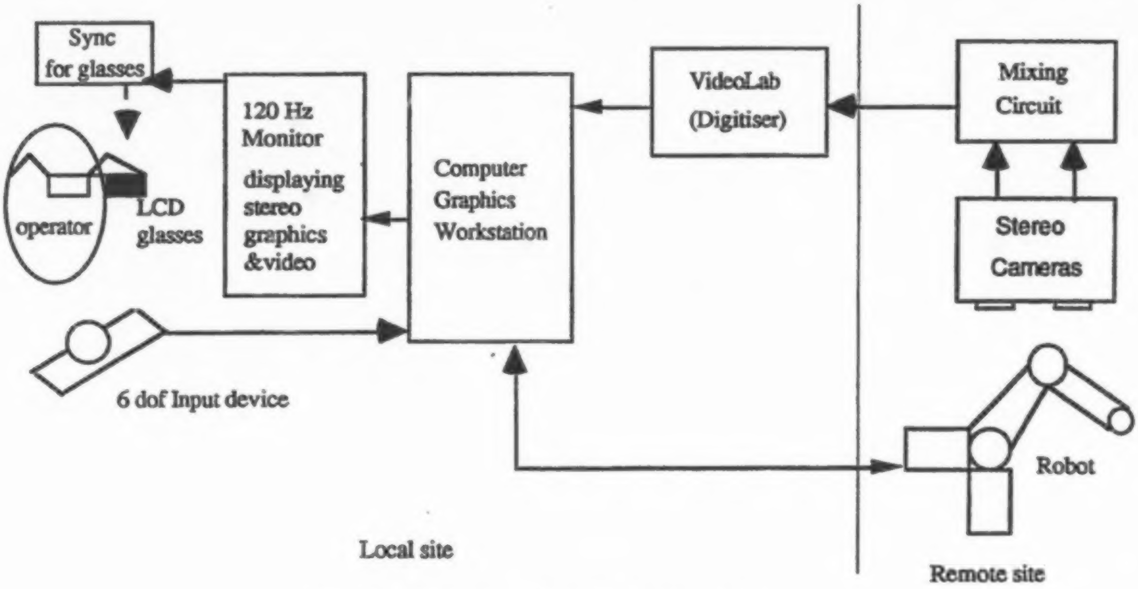


Figure 2: Illustration of system configuration

it overlaps the real robot, to prevent obscuring the remote robot view. When not in motion, it retains the veridical graphical model to give a realistic sense of virtual presence. Operating parameters such as joint limit exceedence checking, end effector trajectory, and world model modification are verified during the simulation.

Control of robot can be decided flexibly in response to task related requirements. For example end effector control, joint by joint control, coordinate system control, or movement with end effector aligned in a particular orientation are all possible. The end effector remains constrained in its operational workspace and, if the controlling virtual end point goes outside this range, the link whose joint limit is exceeded changes in colour to inform the operator and restricts the end effector of virtual robot to move out of task space of robot. Even in concurrent control the robot is protected from joint limit excursion.Furthermore the obstacles can be interactively encapsulated and highlighted with *virtual encapsulators* by means of the 6 d.o.f input device. In the current version only the kinematics of the robot have been modeled. It is also not able to correct for any *continuous* movements of the cameras with respect to the environment and the robot.

The overall block diagram of our virtual control system is given in Fig 2. The system uses two matched, genlocked stereoscopically aligned cameras, which send separate video signals into a switching circuit at the remote site. The two video signals are mixed to form an NTSC video signal with odd and even fields corresponding to the left and right camera views respectively. The interlaced video images are captured by a Videolab™, which is part of a SGI Crimson VGX ™ graphic workstation. Stereoscopic displays are generated at 60 Hz, using 120 Hz top-bottom format, and viewing is done with CrystalEyes™ stereoscopic spectacles. The graphics are generated using SGI Inventor 1.0™. A Spaceball™ is the 6 d.o.f controlling device. A CRS Plus M2A™ robot serves as a robotic manipulator. The current implementation of teleoperation is not dependent on higher end graphics involving simulation, rather supported by it. With a smaller graphical machine our solution can still be implemented, although with more limited graphical support and higher operator involvement. We have also endeavoured to make the code portable to machines having lesser graphics capabilities.

Conclusion

Virtual telerobotic control attempts to construct an operating environment suitable for optimum allocation of tasks between the human and machine. The augmented reality based system presented here attempts to address some of the problems associated with operating in unstructured environments in a cost effective manner. By providing the operator a level of control which is substantially higher than conventional low level manual control, we aim to harness such human resources as dealing with novel situations, creativity and problem solving.

Acknowledgments

This project is supported by the Manufacturing Research Corporation of Ontario (MRCO), as well as by contract W7711-7-7009/01-SE with the Defence and Civil Institute of Environmental Medicine.

References

Bejczy, A.K., Kim, W.S., Venema, S.C., "The phantom robot: Predictive displays for teleoperation with time delays", *IEEE International Conference on Robotics and Automation 1990*, Vol. 1, pg 546.

Browse, R.A. & Little, S., "The effectiveness of real time graphic simulation in telerobotics", *IEEE International Conference on Robotics and Automation*, 1991, Vol. 2, 895-897.

Brunner, B., Hirzinger, G., Landzettel, K. & Heindl, J., "Multisensory shared autonomy and tele-sensor-programming – Key issues in the space robot technology experiment ROTEX", *Proc. IEEE/RSJ Int'l Conf. on Intelligent Robots and Systems, Yokohama*, 1993, 2123-2139.

Burtnyk, N. & Greenspan, M.A., "Supervised autonomy – partitioning telerobotic responsibilities between human and machine", *International Conference on Intelligent Teleoperation*, Nov, 1991, 34-42.

Cannon, D. & Leifer, L.J., "Point and direct telerobotics", *International Conference on Intelligent Teleoperation*, Nov, 1991, 95-106.

Christensen, B., Drotnig, W. & Thunborg, S., "Graphical model based control of intelligent robots", *IEEE Int'l Conference on Robotics & Automation*, 1991, Vol. 2, 1069-1075.

Drascic, D., "Skill acquisition and task performance in teleoperation using monoscopic and stereoscopic video remote viewing", *Proc. Human Factors Society 35th Annual Meeting-1991*, 1367-1371.

Drascic, D. & Grodski, J.J., "Using stereoscopic video for defence teleoperation", *SPIE Vol 1915 Stereoscopic Displays and Applications IV*, 1993, 1-12.

Drascic, D. & Milgram, P., "Positioning accuracy of a virtual stereographic pointer in a real stereoscopic video world". *Proc. SPIE 1457, Stereoscopic Displays and Applications II*, 1991.

Ince, I., Bryant, K. & Brooks, T., "Virtuality and reality: A video/graphics environment for teleoperation," *IEEE Int' Conference on Robotics and Automation*, 1991, Vol. 2, 1083-1089.

Kim, W.S., Tendick, F. & Stark, L.W., "Visual enhancements in pick and place tasks: Human operators controlling a simulated cylindrical manipulator", *IEEE J. Robotics and Automation*, Vol. RA-3, No. 5, Oct '87, 418-425.

Kim, W.S. & Bejczy, A.K., "Graphics displays for operator aiding in telemanipulation", *IEEE International Conference on Robotics and Automation*, 1991, Vol. 2, 1059-1068.

Kim, W.S., Schenker, P.S., Bejczy, A.K. & Hayati, S., "Advanced graphics interfaces for telerobotic servicing and inspection", *Proc. IEEE/RSJ Int'l Conf. on Intelligent Robots and Systems, Yokohama*, 1993, 303-309.

Merritt, J.O., "Often-overlooked advantages of 3-D displays", *Proc. SPIE 902: Three Dimensional Imaging and Remote Sensing*, 1988.

Milgram, P., Drascic, D. & Grodski, J., "Enhancement of 3-D video displays by means of superimposed stereo-graphics", *Proc. Human Factors Society 35th Annual Meeting*, 1991.

Milgram, P. & Krüger, M, "Adaptation effects in stereo due to on-line changes in camera configuration", *Proc. SPIE Vol 1669-13, Stereoscopic Displays and Applications III*, 1992.

Milgram, P. Zhai, S., Drascic, D. & Grodski, J., "Applications of augmented reality for human-robot communication", *Proc. IEEE/RSJ Int'l Conf. on Intelligent Robots & Systems (IROS)*, Yokohama, July, 1993.

Oxenburg, S.C. & Landell, B.P., "Geometric database maintenance using CCTV cameras and overlay graphics", *SPIE Vol. 1006 Space Station Automation IV 1988*, 115-123.

Pepper, R.L., Smith, D.C. & Cole, R.E., "Stereo improves operator performance under degraded viewing conditions", *Optical Engineering*, Vol. 20, no 4, July/Aug '81, 579-585.

Rocheleau, D.N. & Crane, C.D., "Development of a graphical interface for robotic operation in a hazardous environment", *IEEE Int'l Conf. Robotics & Automation*, 1991, 1077-1081.

Ruffo, K. & Milgram, P., "Effect of stereographic + stereovideo "tether" enhancement for a peg-in-hole task", *Proc. IEEE Annual Conf. on Systems, Man & Cybernetics*, 1992.

Schebor, F. S. & Turney, J.L., "Realistic and consistent telerobotic simulation", *IEEE International Conference on Robotics and Automation 1991*, 889-894.

Sheridan, T.B., *Telerobotics, Automation, and Human Supervisory Control*. MIT Press, 1992.

Spofford, J.R., Garcia, K.D., Gatrell, L.B., "Machine Vision Augmented Displays for Teleoperation", *IEEE International Conference on Systems, Man and Cybernetics*, 1991, 19-23.

Zhai, S. & Milgram, P., "Human-robot synergism and virtual telerobotic control", *Proc. Annual Meeting of Human Factors Association of Canada*, Oct. 1992.

Zhai, S. & Milgram, P., "A telerobotic virtual control system", *Proc. SPIE 1612, Cooperative Intelligent Robotics in Space II, Boston*, 1991.

90C221

Defence Teleoperation and Stereoscopic Video

David Drascic
Human Engineering Research & Consulting
241 Logan Avenue, Toronto, Ontario, Canada M4M 2N2
drascic@ie.utoronto.ca

Julius J. Grodski
Defence and Civil Institute of Environmental Medicine
PO Box 2000, Downsview, Ontario, Canada M3M 3B9
jul@dretor.dciem.dnd.ca

#149668

Abstract

This paper examines whether the potential benefits outweigh the expected costs of using stereoscopic video (SV) instead of monoscopic video (MV) for hazardous materials teleoperation. The first part presents the various benefits ascribed to SV, discusses why these benefits should apply to teleoperation, and outlines the expected costs. The second part presents two laboratory experiments showing that using SV can dramatically decrease learning time and improve performance. The third part presents two operations-like experiments conducted with trained telerobot operators of a variety of skill levels; it demonstrates that operators strongly prefer SV, consider it significantly better for most teleoperation tasks, and rate SV to be more useful and more comfortable to use than MV.

Introduction

There are many tasks which are hazardous to human life that could be accomplished remotely using telerobotic manipulators. The need to work with hazardous materials, whether nuclear, chemical, or military, and in extreme and hostile environments, is becoming more and more relevant to modern society and its economy. Efficient deployment of teleoperated and telemanaged robots will be essential for future operations in all of these areas. Currently, most telerobotic systems are completely manual, requiring constant human attention for both high level tasks, such as image interpretation and decision making, and low level tasks, such as driving from one location to another and simple obstacle avoidance. Work is progressing in off-loading some of these low level tasks from the human to the machine, so that the telerobot is semi-autonomous (Drascic et al, 1993, Zhai & Milgram, 1991). In more distant future, fully autonomous robots will be available that can carry out complex operations in unstructured environments without human intervention.

The goal of this work is to examine whether the benefits of using stereoscopic video (SV) for defence-related teleoperation outweigh the costs involved. The particular task being studied is that of *Explosive Ordnance Disposal* (EOD), more commonly known as bomb disposal. This includes handling and disarming both military explosives and *improvised explosive devices* (IEDs). This paper begins with an analysis of the various benefits ascribed to SV, and an outline of the associated costs. We then discuss the relevance of laboratory-based experimental results to real world teleoperation, and present a methodology that stresses the importance of expert evaluation as a robust and powerful analytic tool for field trials. Finally, we present two experiments that used trained operators of a variety of skill levels, seeking confirmation that the expected benefits of SV apply to real world operations.

Limitations of Monoscopic Video

The main source of information to the operator in most existing telerobots is one or more monoscopic video (MV) displays. MV displays eliminate all binocular depth cues (i.e., eye convergence and retinal disparity), as well as some monocular depth cues (i.e. texture gradient). The loss of these important depth cues results in situations where the location of objects in the remote scene is ambiguous. Motion parallax or multiple views can sometimes resolve these ambiguities, but operating conditions typically render these options unfeasible.

A related problem is the difficulty in estimating sizes with an MV system. It is difficult to determine whether an obstacle is too steep to climb, a depression is deep enough to present a hazard, or a door is wide enough for the telerobot to pass through. These deficiencies led to one British study (Robinson, 1984) reporting that standard MV systems made bomb squad personnel reluctant to use their remote manipulator.

Benefits of Stereoscopic Displays

Stereoscopic displays can be used to overcome the limitations of monoscopic displays. Careful implementation is required, where symmetry, precise camera matching (luminance, chrominance, focal length, etc.), and precise camera system alignment must be achieved and maintained. (Starks, 1992, Beldie & Kost, 1991, McGovern, 1987) When implemented properly, SV provides an immediate and compelling sense of depth, which can greatly simplify teleoperation tasks, particularly those requiring delicate manipulation. The literature reports a variety of benefits in using SV displays, including:

- *Faster, more accurate perception of the remote scene.* SV uses the natural depth perception of the operator to convey spatial information.
- *Enhanced detection of slopes and depressions.* With MV, the presence and inclination of a slope or ditch can be difficult to determine. What in MV may look safe is revealed by SV to be hazardous (Merritt, 1991).
- *Enhanced object recognition and detection.* SV presents a three dimensional image, not a flat two dimensional image. With the extra depth information it is often much easier to identify an object; camouflaged objects invisible with MV may be clearly visible in SV (Merritt, 1988).
- *Visual noise filtering.* Visual noise, such as sediment floating in water or dust in a mine, is filtered out by the human binocular vision system, allowing a much clearer image than is possible with MV (Pastoor & Beldie, 1989, Pepper et al, 1981).
- *Faster learning.* A previous study (Drascic, 1991), found that novices using SV perform considerably better at manipulation tasks than those using MV.
- *Faster task performance with fewer errors.* Many studies report better performance with SV under a wide variety of tasks and operating conditions. One study (Drascic, 1991) showed that for high-precision (difficult) tasks, and even for low precision (simple) tasks which constantly vary, subjects perform manipulation tasks faster and with fewer errors when using SV. With sufficient repetition at a simple unvarying task, however, the benefits of SV fade.

Particular Benefits for Defence Teleoperation

Defence related teleoperation tasks, such as bomb disposal and hazardous materials management, are characterised by the need to be conducted successfully on the first attempt while operating in an unpredictable and changing environment. The luxury of repeating a task several times is usually not an option. Thus it is reasonable to expect the benefits of SV to be significant and important even for very simple tasks. For difficult tasks, SV will likely mean the difference between success and failure.

Two potential benefits of using SV for defence teleoperation not previously discussed include:

- *A greater likelihood that the telerobot will be used.* EOD operations constitute a particularly important class of teleoperation tasks performed by the military as well as police forces. Munitions technicians have the responsibility of ensuring that the bomb is dealt with in as safe a manner as possible. If there is any doubt that the task can be accomplished safely using the telerobot, the technicians will not use it, and will neutralise the bomb manually. This puts the technician at risk. If SV allows the operators to use the telerobot more skilfully, they may be more willing to use it, and thus reduce the risk to themselves.
- *A greater range of possible tasks.* Telerobots are often limited in the range of tasks for which they can be used by their design, and also to some extent by the limitations of the MV display. For example, without SV, it is very difficult to use the telerobot in bomb disposal to lower an X-Ray plate behind an IED. So, this task is not attempted remotely. However, this task is considerably easier with SV, and it may be possible for operators to add this task to the repertoire possible with the telerobot.

Costs of Stereoscopic Video for Teleoperation

While the benefits of using SV for teleoperation appear quite convincing, it is important to consider at what price these benefits come. The costs associated with replacing an existing MV set-up with SV can be broken down into hardware, operational, user, and social costs.

Hardware Costs

- *Equipment Costs* can range from US$3,000 (for Alternating Field NTSC SV) to US$18,000 (for Alternating Field 120-Hz SV) to US$35,000 (for colour autostereoscopic displays) for the SV electronics alone. The second camera, cables, mounts, and perhaps replacement monitors can add considerably more expense. On the other hand, prices for such systems have fallen dramatically in the last few years, and will likely continue to do so.
- *Development Costs* include (i) research to determine an appropriate camera configuration for the range of tasks planned; (ii) development of a suitable mount for the cameras, which may need to be either manually or remotely adjustable; (iii) mounting hardware to attach the extra SV components to the telerobot; and (iv) interface electronics to integrate the SV system with the telerobot.
- *Start-up Costs* include manufacturing custom parts, installation of the SV system, training of maintenance people, camera matching and alignment, and development of new operating procedures.

Operational Costs

- Many teleoperation tasks require multiple viewers of the display. Some SV displays, particularly autostereoscopic ones, are inconvenient for multiple viewers.

Additional Maintenance may be required, since there are more components to break down.

Camera Alignment & Matching is very critical. This requires either extra engineering time to ensure that this task need only be done once, or extra operational time to allow this to be done in the field whenever required, or as part of regular maintenance.

Lack of a Zoom Lens for stereoscopic systems can be a distinct drawback in many environments. While work is progressing (Scheiwiller et al, 1991), such systems are not yet commercially available.

User Costs

Acceptance: operators who have been doing their job for years using MV may be reluctant to use SV.

Training: some additional training will be needed, in order to allow the user to become accustomed to using an SV system, to learn how to use the telerobot with it, and to learn the new operating procedures that accompany it. This should be considerably less than the time needed to master and maintain the skills needed to use a monoscopic display, however. (Clapp, 1986)

Comfort: while the development of autostereoscopic displays (i.e. those that do not require the use of special glasses to see the display) is progressing (Eichenlaub, 1993), they remain very expensive, and impose considerable limits on viewer's head position. Thus, for the foreseeable future, viewing glasses will be required with the non-autostereoscopic systems. Glasses may be cumbersome and unpleasant for the operators, particularly for those who wear eyeglasses.

Hitherto Unforeseen Side-Effects of SV may remain hidden until actual field work is attempted. Field use is considerably different from laboratory use, and problems that may be insignificant in the laboratory may jeopardise the success of a mission in the field.

Social Costs

The estimate of the number of *stereo blind* people is generally accepted to be approximately 10% of the population, with another 10% suffering some deficiency (Starks, 1992). Should these people be screened for and excluded from teleoperation tasks?

These costs are clearly not inconsequential, and may outweigh the benefits of SV, depending on the particular circumstances of the teleoperation task.

Balancing the Costs and Benefits

The benefits of SV expected for defence teleoperation include faster and more accurate performance with fewer errors, with considerably less training required over an operator's career. The cost of failure in a bomb disposal task or hazardous materials handling tasks can be very high, in terms of damage to the telerobot, the environment, and life around the site. Furthermore, the expensive training and practice time for the telerobot operators to acquire and maintain sufficient skill at interpreting the monoscopic cues of an MV display will likely be considerably reduced with SV. If these expectations prove valid in actual field use, it would be prudent to invest in SV for defence teleoperation.

The Experiments

Two experiments were conducted using trained telerobot operators with a variety of skill levels, seeking confirmation that the expected benefits of SV apply to real world field operations. The first experiment was conducted under field-like conditions with typical operators, who as Munitions Technicians have training but little daily experience using the telerobot. The second experiment was conducted under more controlled conditions with expert telerobot operators.

Human-Machine System Performance Evaluation

All of the benefits of SV described above were obtained either by theoretical analysis or by experiments conducted in a laboratory under limited and controlled laboratory conditions. The experimental tasks used were limited in scope and for the most part highly repeatable. Real EOD tasks are broad in scope and unique in circumstance: no two field incidents are ever identical. The question arises of whether or not it is reasonable to expect laboratory results to be relevant for actual field conditions. These laboratory experiments were designed to examine some of the skills operators need to exercise in carrying out teleoperation tasks. While these skills comprise only a portion of the total effort, their contribution is very important for the successful completion of the task. For example, a peg-in-hole task is not something typically done by EOD operators in the field, but many of the same skills are required for various portions of an EOD event. Therefore, it is reasonable to expect that the benefits of SV found in the laboratory will pertain the field.

In order to properly evaluate the benefits of SV for EOD, it is important to avoid the two typical flaws committed by many researchers conducting field trials (Hennessy, 1990). First, there is an assumption that people behave in a regular, repeatable, predictable, and explicable manner, much like machines. While tight laboratory controls may allow this assumption to stand, it fails completely in a field environment where events are unpredictable and operators must draw on resources and experience not codified in advance by the experimenter.

The second flaw occurs when an experimenter attempts to use laboratory techniques in field situations (Hennessy, 1990). Standard statistical methods were developed to study highly repeatable natural and mechanical systems that did *not* have an active and unpredictable human intelligence. When analysing human-machine systems, laboratory techniques require tight controls and a strictly limited scope. They are insensitive to variations in circumstances and the unpredictability of the field and most importantly, they ignore situation and context when attempting to understand system behaviour. Attempting to use these techniques to measure performance causes the largest part of the data variability to remain inexplicable. These techniques cannot isolate various contributing factors, and are not comprehensive: they can miss important factors.

The proper way to appraise performance in the field is to use the assessment of human experts with experience in the task being studied (Hennessy, 1990). The expert ratings can then be subjected to statistical analysis to test for consistency in response. If the experts are in agreement, their opinions and interpretations are a reliable and powerful analytic tool. In these experiments, the human experts referred to are the EOD technicians themselves. Previous investigations into SV have used this approach, both directly by gathering human ratings (Gorski, 1992), and indirectly, by observing which system is used by an operator when both are available (Spain & Holzhausen, 1991, Dumbreck et al, 1990). Our experiments use the direct approach of gathering operator ratings.

Stereoscopic Video System Configuration

The field of view of the SV system was matched to that of the standard black and white camera with which the telerobots were equipped, in order to allow the operators to make as fair a comparison as possible between the SV and the MV systems. We used Panasonic GP-KS102 1/3 CCD cameras equipped with 6 mm c-mount lenses. The standard displays of the telerobots used, Hitachi 9 low resolution monochrome displays. In order to give suitable stereoacuity, the cameras were separated by approximately 12 cm, converging at the tip of the manipulator arm, approximately 80 cm from the cameras.

The SV system used the *alternating field* encoding approach for transmission and display. (Milgram et al, 1990) The video field alternator used was developed by Dr. Paul Milgram of the University of Toronto and the authors. A 3D-TV StereoDriver Model 1000 was used to control Nintendo (3D-TV Model N) liquid crystal shuttering spectacles (Starks, 1993).

Experiment 1: Operators in Field Conditions

Subjects

The first experiment had 8 subjects of self-rated moderate to good skill level with the telerobot. This group of seven men and one woman were munitions technicians, with various degrees of training and experience with the telerobot. They constituted a typical pool of operators. As is typical for field trials, this experiment was conducted under very strict time, personnel, and equipment limitations.

Method

A full factorial within-subjects design was used to ensure that all subjects experienced all conditions, and that the exposure was counter-balanced. Subjects performed all tasks in both MV and SV conditions, in order to be able to make a fair comparison. Each subject participated in two experimental sessions of approximately two hours on separate days. They would use either MV or SV for the first session, and the other for the second session. In each session they would perform each of the tasks described below once.

The two telerobots were used in this experiment were considerably different, due to circumstances typical of field trials. We chose to use the older, considerably less dependable telerobot for the SV display in order to bias this experiment *against* the expected benefits, using a well-behaved new one for the MV display. In this way, if the expected benefits were found, it would indicate that they were robust to such problems. As reported by our operators, the older robot was the direct cause a significantly higher number of performance errors with SV than might otherwise have been expected. *While standard laboratory experimental analysis would be ruined by this uncontrollable influence, the field trial approach used was able to isolate this factor from the effects of the video system. This technique is clearly a more powerful, robust, and reliable analysis method for field situations.*

Tasks

Three tasks were chosen to represent many of the skills required to successfully control the telerobot in the field. They were based on EOD procedures, but did not adhere strictly to them.

Manipulation: The operators used the telerobot to pick up 8 blocks, one at a time, without knocking them over and place them in a bin. This task was performed once at the beginning of each session using direct view in order to serve as training for the operators in the use of the telerobot, and then once again under each remote viewing condition.

Weapon Positioning: This task required the operators to precisely position an EOD weapon. Errors consisted of any unintentional contact with the environment, and particularly with an IED (improvised explosive device). This task was repeated five times by each subject using both MV and SV.

X-Ray: This task required the operators to lower an x-ray plate, approximately 25 cm high by 40 cm wide, between two briefcases separated by approximately 15 cm. The plate was attached to a boom on the telerobot, and could swing freely. This task was repeated five times by each subject in both video conditions.

Expectations

Given the previous work showing that SV has considerable advantages for almost all teleoperation tasks, we expected that the EOD operators would respond favourably to it, and would feel more capable of controlling the telerobot. We hypothesised that the subjects would rate both systems equally comfortable. Since subjects were instructed to perform the tasks according to standard procedure, there was no time pressure, and so we expected that there would be no significant difference in task time between the MV and SV. Prior to the experiment, we hypothesised that there would be fewer performance errors with the SV system. However, given the difference between the two telerobots used in the trials, this hypothesis was discarded.

Experiment 2: Experts, Controlled Conditions

Subjects

The second pool of 8 subjects of self-rated good to expert skill levels were experts at EOD. Several were instructors.

Method

A partial-factorial within-subject design was used. The subjects participated in a single experimental session. The identification task described below was performed twice at the beginning of the session, once with MV and once with SV. The other three tasks were each performed once as a set with MV, then as a set with SV, forming one block of two sets. Each block was repeated 5 times. The entire session took between 3 and 4 hours to complete. A single telerobot, equipped with a switch-able MV/SV video system as described above, was used for both video conditions.

Tasks

Identification: The subjects were presented with 4 out of a set of 8 similar IEDs standing on a table. The operators would use the telerobot to examine each IED to determine whether or not it was complete and would function. If they said it would not function, they had to identify why (e.g. missing part, broken wire).

Manipulation: The subjects had to pick up four blocks from one shelf and transport them to a second one, placing them standing up on the edge. The blocks and the shelves were speckled grey, which served to camouflage them and increase the task difficulty.

Weapon Positioning: The subjects were required to precisely position a weapon against an IED. Standard procedure, developed to assist those using MV, was used: a loop of flimsy tape was attached to the end of the weapon so that it projected forward . When the operators saw the tape move, they knew they had attained the correct distance. Once in position, the subjects were then required to estimate the angle of the weapon with respect to the IED. This was compared with the actual angle to get a measure of the perception of depth of the subjects.

X-Ray: This task required the operators to lower an x-ray plate between two briefcases separated by approximately 15 cm. Errors included touching either of the briefcases.

Measures

The main measures in this experiment were the rating by the operators regarding the usability and comfort of the MV and SV systems, and a rating of preference. Other measures include task completion time, number of errors, and subject ratings of performance and confidence.

Expectations

It was expected that these subjects, being more skilled than the previous group, might show less of a preference for SV, but would still prefer it. Again, we hypothesised that there would be no difference in comfort rating. The repetition of the tasks under both video conditions 5 times imposed some measure of laboratory control on the experiment, and we hypothesised that tasks would be performed more accurately and with fewer errors with SV.

Observations

The subject ratings of Experiments One and Two are presented in Figures 1 through 4 and discussed below. In summary, all of the operators responded very positively to the SV system. They expressed a strong desire to have the entire telerobot fleet equipped with SV. They reported that SV was more useful than MV, and at least as comfortable to use. Statistical analysis of the responses indicate a high degree of consistency in most measures (Table 1). There were no significant effects of order of exposure: those who used SV in their first session gave essentially the same ratings as those who used MV.

As expected, there were no significant differences in the task time or error rates for the different tasks in Experiment One. However, performance advantages for SV were found in Experiment Two. These are summarised in Table 2.

Operator Assessment of SV for EOD

The operators in both experiments were asked a number of questions to determine their objective evaluation of the two video systems. The operators showed the typical naive trends of tending toward the mean and avoiding the extremities, no matter how much these extremes may be warranted. The questions that were asked and their corresponding answers are presented below. A summary of the results is given in Table 1.

Table 1: Mean Operator Ratings

Mean Ratings	Usability		Comfort	
	Exp. 1	Exp. 2	Exp. 1	Exp. 2
MV	4.625	3.75	4.75	5.062
SV	6.25	5.625	6.00	4.812
Paired 1-tail	t(7)=3.05 p=.009	y(7)=9.1 p<.0001	p(7)=2.8 p=.014	p(7)=.68 p=.258

Question 1: "How would you rate the video system you used with respect to usability, i.e. how well it helped you get the job done, where 1 is terrible, 4 is acceptable, and 7 is excellent?"

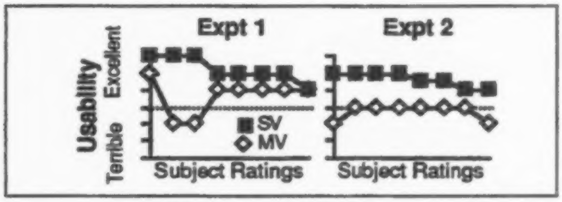


FIGURE 1: Usability Ratings

Question 2: "How would you rate the video system you used with respect to comfort, where 1 is terrible, 4 is acceptable, and 7 is excellent?"

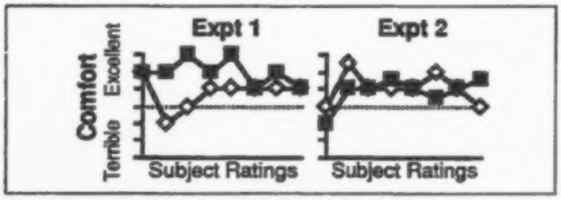


FIGURE 2: Comfort Ratings

Question 3: "If you had to choose between having either an MV or a SV system, but not both, how would you rate your decision, where 1 means you strongly prefer MV, 4 means you have no preference, and 7 means you strongly prefer SV?"

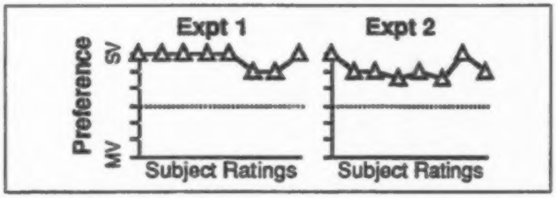


FIGURE 3: Forced Choice Between MV and SV

Question 4: "If the telerobot was equipped with both MV and SV system which you could toggle between with the touch of a switch, how much of the time do you think you would have the telerobot in the SV mode, where 1 is never, 4 is half the time, and 7 is always?"

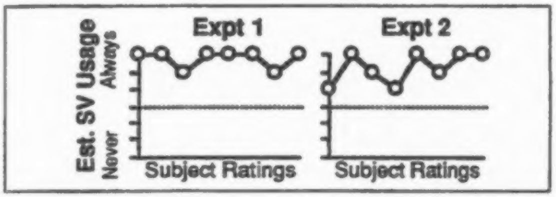


FIGURE 4: Estimated SV Usage if both are available

Discussion

Usability (Figure 1) is an imprecise term: it implies how well something affords, or facilitates, its intended use. It is not quantifiable per se, and cannot be used with traditional automatic data gathering techniques. However, it is a term that all the operators understood clearly. The operators that participated in Experiment 1 are highly trained munitions technicians, but since the EOD telerobot is rarely used by them, few had much day-to-day experience. The expert operators of Experiment 2 were instructors, with considerably more experience with the telerobot. It is clear from the ratings of both experiments that SV was considered much more useful for the EOD tasks than was MV, regardless of the skill level of the operator.

The comfort rating (see Figure 2) in Experiment 1 that indicated that SV was *more* comfortable than the similar MV system was unexpected. The explanation arose in the debriefing interviews, when several subjects explained that it was easier to see things with the SV display, so they spend much less time peering closely at it than they had at the MV display. That the comfort ratings in Experiment 2 showed no difference was anticipated by several subjects in Experiment 1, who indicated that the flicker of the SV was acceptable for the relatively short exposure of 1 hour, but could be less acceptable for a longer exposure. In situations requiring extended operation, the results of Experiment

Table 2: Experiment Two Task Results

TASK	MV	SV	SV=%MV	COMMENTS
Identification Task	65% correct	75% correct	115%, minor improvement	t(18) = 0.802, p=0.433. *This task had fewer repetitions than others*
Manipulation Task	66% success	80% success	121%, significant improvement	F(1,7) = 8.795 p = 0.021
Weapon Positioning Task	**MV - Avg. Absolute** Error in Angle Estimate	**SV - Avg. Absolute** Error in Angle Estimate	**Results of SV expressed as a percentage of MV**	**Operator estimate of weapon angle improved by up to 33% with SV**
Target on the floor:	13.8	9.2	67%, large improvement	*steep angle required, coupled with steep angle of camera and telerobot.*
Target on a table:	5.8	6.2	107%, minor degradation due to occluded view	*no angle required. Unfortunate camera position meant that the right eye for SV was completely obscured by the near end of weapon.*
Target on a shelf:	20.5	16.3	80%, some improvement	*steep angle required, with moderate angle of camera and telerobot.*
Mean	13.4	10.6	85%, some improvement overall	F(1,4) = 4.601, p = 0.015. *Benefits are task dependent, subjects tend to underestimate angle with MV.*
X-Ray Plate Task	**MV Avg. Dist.**	**SV Avg. Dist**	**SV=%MV**	**With SV, Plate Distance 47% closer, Plate Angle 22% smaller**
Distance of Plate from target (smaller is better)	6.6cm	3.5cm	53%, considerable improvement	t(78) = 4.390 p = 0.000
Angle (difference between left and right edge)	2.3cm	1.8cm	78%, slight improvement	t(78) = 1.400 p = 0.165

2 suggest that the operators will suffer no more discomfort with SV than they would with MV.

Debriefing interviews revealed further insight into the issues explored above. In response to the question ∍Did you notice/suffer any headaches that you would attribute to the SV system? , all 16 subjects said ∍No.

In response to the question ∍Did you notice any visual fatigue you would attribute to the SV system? :

8 said no, one adding ∍I thought I would, but didn t.
3 said the flicker was distracting at first
2 said ∍a little at first
2 said ∍a little
1 said ∍more than MV .

All subjects were asked how they would improve the displays. All said they would get rid of the flicker and the goggles. So while the ratings say that the flicker of a 60 Hz alternating field SV display *is* acceptable for both casual and extended use, their comments suggest that a flicker-free display would be preferred by the users.

When forced to choose between MV and SV (Figure 3), all subjects expressed a strong preference for the SV. When presented with the option of having both available and being able to switch freely between them (Figure 4), all operators said they would use SV most or all of the time. In conjunction with the usability and comfort ratings above, these ratings indicate that SV would be of considerable advantage to EOD teleoperation.

Given the nature of field trials and the inappropriateness of laboratory statistical techniques for them, it is neither surprising nor a failing of the technique used that no differences were found in Experiment 1 for the ∍traditional measures of performance (i.e. task time and error rate), particularly in light of the confounding of video system with quality of telerobot. That they were found at all in Experiment 2 is an indication of how strong the benefits must be. Experiment 2 was designed to be disruptive, with no task condition re-occurring until all others had passed, and with the video system switching between MV and SV throughout, in order to prevent short term learning. Despite this, several important performance advantages to SV were found (Table 1). The estimate of weapon position, which comes from more accurately perceived slopes with SV, is particularly important: the angle with which the weapon is aimed is critical for the success of the mission. If aimed incorrectly, the weapon could accidentally detonate the IED, instead of disabling it.

Conclusion

In order to evaluate the potential benefits of using stereoscopic video (SV) rather than monoscopic video (MV) for defence teleoperation, two experiments were conducted, where operator ratings were used as the main measure of system performance. These experiments found that both naive and expert operators expressed unanimous acceptance and preference for SV over MV, and agreed that SV was superior to MV for all tasks requiring precision operation. They also rated SV more useful and more comfortable than MV.

The performance benefits of SV found in these experiments include better object identification, improved manipulation, improved weapon positioning, and improved perception of the remote environment. It was seen that SV can be used to extend the range of tasks for which the EOD telerobot can be used.

The ratings of the trained telerobot operators, and the performance advantages found, indicate that SV would be of considerable benefit to defence teleoperation.

Acknowledgement

The work described herein was done under contract W7711-0-7114/01-XSE with Supply and Services Canada for the Defence and Civil Institute of Environmental Medicine. The authors would like to thank Captain John Eng and the Canadian Forces School of Electrical and Mechanical Engineering at CFB Borden for their participation and support in this work.

References

Beldie, I P, Kost, B, зLuminance asymmetry in stereo TV images, SPIE Vol. 1457 Stereoscopic Displays and Applications II, 242-247, 1991

Clapp, R E, "Stereoscopic displays and the human dual visual system," SPIE Vol. 624 Advances in Display Technologies VI, 41-52, 1986

Drascic, D, "Skill Acquisition and Task Performance in Teleoperation using Monoscopic and Stereoscopic Remote Viewing," Proceedings of the Human Factors Society 35th Annual Meeting, 1367-1371, 1991

Drascic, D, Grodski, J, Milgram, P, Wong, P, Ruffo, K, Zhai, S, "ARGOS: A Display System for Augmenting Reality," in the Video Proceedings of INTERCHI ≅93 Conference on Human Factors in Computing Systems, Amsterdam, 1993

Dumbreck, A A, Abel, E, Murphy, S P, з3-D TV System for remote handling: development and evaluation, SPIE Vol. 1256 Stereoscopic Displays and Applications, 226-236, 1990

Eichenlaub, J B, "Developments in autostereoscopic technologies at Dimension Technologies, Inc.," SPIE Vol. 1915 Stereoscopic Displays and Applications IV, 1993

Gorski, A M, "User evaluation of a stereoscopic display for space-training applications," SPIE Vol. 1669 Stereoscopic Displays and Applications III, 236-243, 1992

Hennessy, R T, "Practical Human Performance Testing and Evaluation," Chapter 15 of MANPRINT: An Approach to Systems Integration, H R Booher, ed., Van Nostrand Reinhold, New York, 1990

McGovern, D E, "Current development of needs in the control of teleoperated vehicles," SANDIA National Laboratories Report SAND87-0646 UC-15, Albuquerque, New Mexico, 1987

Merritt, J O, "Often-overlooked advantages of 3-D displays," SPIE Vol. 902 Three-Dimensional Imaging and Remote Sensing Imaging, 46-47, 1988

Merritt, J O, "Perceptual training with cues for hazard detection in off-road driving," SPIE Vol. 1457 Stereoscopic Displays and Applications II, 133-138, 1991

Milgram, P, Drascic, D, Grodski, J, "A Virtual Stereographic Pointer for a Real Three Dimensional World," INTERACT ≅90, Third IFIP Conference on Human Computer Interaction, Cambridge, 1990

Pastoor, S, Beldie, I P, "Subjective assessments of dynamic visual noise interference in 3D TV pictures," Proceedings of the Society of Information Display, 30(3), 211-215, 1989

Pepper, R L, Smith, D C, Cole, R E, "Stereo TV improves operator performance under degraded visibility conditions", Optical Engineering, 20(4), 579-585, July/Aug. 1981

Robinson, M., "Remote control vehicle guidance using stereoscopic displays," Proceedings of the Annual Meetings of the Human Factors Society, 1984

Scheiwiller, P M, Murphy, S P, Dumbreck, A A, "A compact zoom lens for stereoscopic television," SPIE Vol. 1457 Stereoscopic Displays and Applications II, 2-8, 1991

Spain, E H, Holzhausen, K P, зStereoscopic versus Orthogonal View Displays for Performance of a Remote Manipulation Task," SPIE Vol. 1457 Stereoscopic Displays and Applications II, 103-110, 1991

Starks, M, "Low cost universal stereoscopic virtual reality interfaces," SPIE Vol. 1915 Stereoscopic Displays and Applications IV , 1993

Starks, M, "Stereoscopic Video and the quest for virtual reality," SPIE Vol. 1669 Stereoscopic Displays and Applications III, 216-227, 1992

Zhai, S, Milgram, P, зA telerobotic virtual control system, SPIE Vol. 1612 Co-operative Intelligent Robotics in Space II, 1991

Operator-robot interaction for task level telerobotics

Paul Freedman*
Centre de recherche informatique de Montréal
1801, avenue McGill College, Suite 800
Montréal QC H3A 2N4
freedman@crim.ca

#149669

Abstract

The ubiquity of distributed microprocessor-based control and computer networking has dramatically changed the design of computerized control systems including telerobotic systems. To an increasing extent, control is mediated by the operator-robot interaction. In this paper, we argue that reliable *system-level behavior must therefore mean, in part, the careful* engineering *of this operator-robot interaction. But in order to define and verify associated safety properties, some kind of formal modelling and analysis is required.*

1 Introduction

Recently, telerobotics for intervention tasks in domains such as forestry eg. [Lawrence 1989], mining eg. [Williams, Jackson and Monteith 1992], and the nuclear industry eg. [Schilling 1992], is gaining increasing importance [Freedman 1993a]. While the robot must necessarily have some degree of decisional and operational "intelligence" in order to adapt in realtime to its work environment[1], it is the human operator who chooses task goals and suggests possible strategies (nominal behaviors) which the robot must thereafter refine according to the particular circumstances present at runtime. This has been called *tele-robotics at the task level* [Alami, Chatila and Freedman 1990].

In contrast to (human) supervisory control [Sheridan 1987], [Sheridan 1992] which emphasizes the autonomous execution of simple tasks by a computerized system, task level telerobotics also means the autonomous decomposition (refinement) of complex tasks into simple sensor-based actions, and their execution monitoring at runtime. In addition, the operator's station must be equipped with complementary programming and monitoring aids in order to make effective use of the robot's additional autonomy. This is the context for the research to be described here.

As computerized control systems (including telerobots) become capable of increasingly autonomous behavior,

> "increasingly human [operators] are placed in the role of system monitors rather than active participants. They are basically asked to remain alert for long periods of time with nothing much to do, and react quickly when needed. This is exactly the sort of task that *machines* do well and humans don't [Spencer 1992].

As a result, we believe that the design of the operator interface must be carefully *engineered* in order to make it possible to establish guarantees about system (operator + robot) behavior in a formal way.

Nonetheless, we submit that human factors issues continue to dominate the research published in the user interface (man-machine) literature eg. [Stassen 1990]. However, there is some indication that the computer science community is beginning to look more carefully at how such interaction might be modelled eg. [Johnson 1992].

The same observation can be made in a telerobotics context where research tends to emphasize human factors aspects eg. [Lumelsky 1991] instead of operator-robot dialogue eg. [DeMori, Bourdeau and Kuhn 1990]. In this paper, we suggest that certain aspects of the operator-robot interaction related to *safety* deserve to be studied in a formal way within the context of discrete event control [Ho 1989]. Here *state* is a mathematical rather than a physical abstraction, and the occurrence of events give rise to changes in state. Time advances in discrete but variable steps, and system dynamics are piece-wise constant.

---

*The financial support of the Natural Sciences and Engineering Research Council is gratefully acknowledged.

[1] This is especially so when the robot is working far from the operator and significant time delay is present in the operator-robot communication. Such would be the case for the telerobotic control of an underwater or mining robot from the surface, or for a planetary robot.

Indeed, we submit that the ubiquity of distributed microprocessor-based control and computer networking has dramatically changed the design of computerized control systems. To an increasing extent, computerized control is mediated by message passing among the controllers and in this sense, control problems are increasingly viewed in terms of designing the interactions between a controller and a controlled process [Ramadge and Wonham 1989]. The controller receives feedback messages from the process and as circumstances dictate, sends command messages to the process in order to change its behavior. In a telerobotic context, the operator plays the role of the controller and the robot that of the controlled process. (The robot in turn plays the role of controller with respect to its work environment.)

We argue here that at the very least, *reliable* task level telerobotics demands that safety properties related to the operator-robot interaction be defined and verified. This necessarily implies that such interaction be subject to formal modelling and analysis.

Of course, the software engineering community has long recognized the importance of modelling computerized systems in a formal way, in order to make precise what must be done, and to represent the evolution of system behavior in time [Laplante 1993]. The combination of the two makes it possible, in a formal way, to establish 'guarantees' about the integrity of the system behavior [Beizer 1990].

Whereas proving program correctness continues to be the subject of much computer science research, the verification of most safety properties is a less ambitious task.

> Almost always, the safety of a critical application does not depend on the total correctness of its control program, but rather on an often small set of properties that the program should fulfill. For instance, the occurrence of a critical situation should raise an alarm within a given delay. From our experience ... these properties seldom depend on numerical relations and computations. [Halbwachs et al. 1991], p. 1314.

In this paper, we suggest how safety properties may be associated with operator-robot interaction for task level telerobotic systems, and then examine how such properties may be verified in a formal way.

2 Safety and liveness

We begin by distinguishing between so-called 'safety' and 'liveness' properties as understood by the computer science community.

Among the earliest definitions are those presented in [Lamport 1977] for describing the correctness of a multi-process (in the computer science sense) program. A safety property describes something 'bad' which must *not* happen, while a liveness property describes something 'good' which *must* happen. An example of a safety property is 'freedom from deadlock', while an example of a liveness property is 'guaranteed service', i.e. a server will eventually respond to all requests from each client [Alpern and Schneider 1985]. Thus, safety is more generally associated with behavioral *invariance*, while liveness is associated with behavioral *inevitability*.

Some additional considerations are presented in [Alpern and Schneider 1985]. Firstly, safety is absolute; one cannot speak of something 'bad' which must not happen some time. Furthermore, it is somehow 'discrete'; if a safety property is violated, then it is violated at an identifiable state. Conversely, liveness may be conditional in the sense that a liveness property may refer to a partial execution; if something 'good' has not yet happened, it may still happen some time later. In addition, a liveness property need not be 'discrete'. For example, it's unclear how the violation of 'guaranteed service' might be associated with a single state since this is a consequence of the system behavior in general.

In previous work [Freedman 1992a], we suggested how liveness properties might be studied within the context of a particular Petri net model of the system behavior [Freedman 1992b]. Here we were concerned with questions such as the following: given a particular action to be performed as selected by a human operator, it is always the case that the system state may be 'driven' to a state in which this action is enabled by a legal series of state transitions ? We also described how such a liveness property may be guaranteed by carefully selecting the initial state of the system.

In this paper, we prefer to distinguish between properties which are 'invariant', i.e. are true for all system states, and properties which are 'non-invariant', i.e. characteristic of just some states following [Sifakis 1982]. Consider, for example, the following safety property: it is always the case that if event e_1 occurs, then event e_2, must occur sometime afterward eg. a pump failure must always raise an alarm.

We submit that such qualifications over events better describe the kinds of safety properties likely to be of operational concern than Lamport's 'safety' properties such as 'freedom from deadlock'. And when events are associated with changes in state [Thistle and Wonham 1986], the verification of such safety properties may be recast as *reachability* questions about the system state space [Freedman 1993b].

In this context, controllability as understood by the control systems community is a kind of 'complete' reachability in the sense that given any two (valid) states, there necessarily exists a path in state space between them. Unfortunately, this is rarely characteristic of most control problems since the closed-loop response typically consists of some repetitive (steady state) behavior and some non-repetitive (transient) behavior associated, perhaps, with initialization or with recovery from errors, operator intervention, etc. Therefore, the system state space is, in general, not strongly connected, and the system is not controllable. Therefore, reachability properties such as reversibility (or the possibility of re-initializing the system state) are not invariant since they are not characteristic of the non-repetitive behavior of the system.

But sensor-based robotics necessarily implies that system dynamics are non-deterministic, i.e. a sensor may return one of many values, each giving rise perhaps to a different robot action, Consider, for example, sonar sensing used for the obstacle avoidance of a mobile robot. At any given time, the sonar 'signature' might indicate no obstacle ahead, or perhaps an obstacle to the left or right. In previous work [Freedman and Malowany 1988], [Freedman 1992b], we suggested that such non-determinism may be studied in a decomposed way by examining each of the possible sub-behaviors, i.e. the consequences of no obstacle ahead, obstacle to the left, etc. But despite such non-determinism in the system dynamics, typical safety properties of interest eg. the robot must not collide with obstacles in the work environment, may also be studied in a decomposed way eg. robot must not collide with obstacles ahead, with obstacles to the left, etc.

3 Modelling and analysis of safety properties

Given a suitable state variable model for describing system dynamics, behavioral analysis may be performed from two complementary perspectives. In the event domain, one is primarily concerned with sequences of events or 'traces'. Conversely, in the state domain, one is primarily concerned with the states themselves and trajectories in state space [Ho 1989].

However, these perspectives are not mutually exclusive; Petri nets (PN) [Murata 1989] may be studied both in terms of states (the Reachability Graph of possible markings of the places) and in terms of events (the firing diagram of the transitions).

To date, most of the research in the computer science literature about investigating safety properties is conducted within the context of temporal logic. Here system behavior is described in terms of propositions whose truth value changes from state to state [Audureau, Enjalbert and Farinas del Cerro 1990].

Given some logical proposition P, four temporal operators are typically defined:

- $\Diamond P$, for P becomes true at some future state (at some future time instant)
- $\Box P$, for P is true in the present state and in all future states
- $\bigcirc P$, for P becomes true in the next state
- $P \, \mathcal{U} \, Q$ for P is true until the first state in which Q is true

Typically, the analysis of temporal logic formulas is performed using 'tableau' methods to decompose the temporal logic formulas into (a) those which describe the present and (b) those which describe the future [Manna and Wolper 1984]. Nodes in the 'model graph' are possible truth assignments to the formulas, and arcs are combinations of (a) and (b). As time is advanced, formulas in (b) are decomposed and new arcs and nodes are created. The analysis continues in this manner until there are no new nodes, i.e. until we have obtained all possible truth assignments to the various temporal logic formulas. Such methods are intrinsically computationally intensive.

Within the framework of temporal logic, safety properties associated with qualifications over sequences of events take the following form:

$$\Box \, (e_1 \Diamond e_2) \tag{1}$$

In words, it is always the case that event e_1 is followed by event e_2.

But in order to verify such a safety property, we must first associate events with changes in state and then recast equation 1 as a temporal qualification over *states*. To keep things simple, we shall associate a *single* event with the change in value of a *single* state variable. Let's now assume that the two events, e_1, e_2 may be defined as follows:

$$e_1 \equiv [x = x_a \mapsto x = x_b] \tag{2}$$

$$e_2 \equiv [x = x_c \mapsto x = x_d] \tag{3}$$

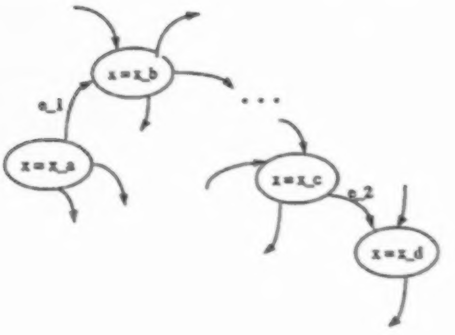


Figure 1: Verifying a safety property in state space.

where x is a state variable taking on values $\{x_a, x_b, x_c, x_d\}$.

In this way, equation 1 may be restated as follows:

$$\Box\left([x = x_a \mapsto x = x_b] \Diamond [x = x_c \mapsto x = x_d]\right) \quad (4)$$

In principle, such a safety property may be verified as follows:

1. Identify the state in which $[x = x_a]$.
2. Follow the arc leading to the (immediate) successor state in which $[x = x_b]$. This is the state transition corresponding to e_1.
3. From this state, *all* outgoing paths must eventually lead to the state in which $[x = x_c]$.
4. If this state has an immediate successor state in which $[x = x_d]$, the safety property is verified.

The situation is best described by the schema shown in Figure 1, corresponding to non-deterministic system dynamics described in terms of the single state variable x.

Of course, such an algorithm presupposes that the system state space is finite and has been previously generated. But since the safety property is only concerned with a (typically small) subset of that state space, it might be wiser to adopt a different kind of analysis which does not require as a first step the generation of the system state space.

In [Freedman 1993b], we demonstrate that when the closed loop behavior is deterministic, safety properties associated with qualifications about sequences of events of the form shown in equation 1, may be recast as reachability questions about markings of an associated PN and investigated *algebraically*. We submit that such an approach is more efficient than the tableau methods associated with temporal logic and the graph-theoretic search methods (such as the algorithm previously presented) for which the state space must be generated as a first step.

More precisely, we begin with a particular kind of PN called Ω defined in [Freedman 1992b]. From this PN, an algebraic description of the system dynamics, i.e. how the net markings is changed by firings the transitions, is developed using the Incidence matrix C and the firing count vector F. Finally, by associating events e_1, e_2 with specific transitions, say t_1, t_2, and by associating markings M_1, M_2 with the states in which these transitions are enabled, it becomes possible to answer, in an algebraic way, the following reachability question:

$$\Delta M = M_2 - M_1^* \equiv CF \quad (5)$$

where M_1^* is the marking *after* firing transition t_1.

In words, if F has an integer solution, then there exists some legal sequence of transition firings to move from marking M_1 in which t_1 is enabled to marking M_2 in which t_2 is enabled.

4 Future work

In this paper, we suggested that the definition and verification of safety properties should be an important part of the *engineering* of operator-robot interaction for task level telerobotic systems.

As part of our current research, we are exploring the 'reactive' model of computerized systems which maintain a 'permanent interaction' with their environments [Harel and Pnueli 1985]. As noted by [Reisig 1992], "reactive systems have been the focus of PNs from the very beginning" p. 25.

In addition, we are also studying the family of *synchronous* programming languages [Benveniste and Berry 1991] for which changes in inputs give rise to changes in outputs in an instantaneous way, as an alternative framework for the formal modelling and analysis of operator-robot dialogue. The synchrony hypothesis makes it possible to ignore the durations of events in order to concentrate on the essentials of the system dynamics. This is also consistent with much work in the discrete event systems literature which often assumes that changes in state take place instantaneously, in order to concentrate on the study of (non-timed) trajectories in state space.

But the typical context for this research is telecommunication systems subject to external constraints imposed by the work environment, and since robots are not telephone systems, we must proceed carefully when examining the the utility of such synchronous languages in a robotic context eg. [Coste-Maniere, Espiau and Simon 1992].

References

Alami, R., Chatila, R., and Freedman, P. 1990. Task Level Teleprogramming for Intervention Robots. In *IARP 1st Workshop on Mobile Robots for Subsea Environments*.

Alpern, B. and Schneider, F. 1985. Defining liveness. *Information Processing Letters*, 21:181–185.

Audureau, E., Enjalbert, P., and Farinas del Cerro, L. 1990. *Logique temporelle*, Masson.

Beizer, B. 1990. *Software Testing Techniques*, Van Nostrand Reinhold, 2nd edition.

Benveniste, A. and Berry, G. 1991. The synchronous approach to reactive and real-time systems. *Proceedings of the IEEE*, 79(9):1270–1282.

Coste-Maniere, E., Espiau, B., and Simon, D. 1992. Reactive objects in a task level open controller. In *IEEE Int. Conf. on Robotics and Automation*, pages 2732–2737, Nice, France.

DeMori, R., Bourdeau, J., and Kuhn, R. 1990. *Workshop on Dialogue Systems*, chapter A probabilistic approach to person-robot dialogue. NATO/ASI.

Freedman, P. 1992a. Modelling the actions of an intervention robot. In *IEEE Int. Conf. on Robotics and Automation*, Nice, France.

Freedman, P. 1992b. A practical Petri net model for the state-oriented modelling of discrete event systems. Technical Report 92/08–05, Centre de recherche informatique de Montreal.

Freedman, P. 1993a. Robotics in the 1990s. *IEEE Canadian Review*, pages 5–8.

Freedman, P. 1993b. Verifying saftey properties of discrete event control systems expressed in temporal logic via algebraic petri net analysis. Technical Report CRIM-93/08-09, Centre de recherche informatique de Montreal.

Freedman, P. and Malowany, A. 1988. The analysis and optimization of repetition within robot workcell sequencing problems. In *IEEE Int. Conf. on Robotics and Automation*, pages 1276–1281, Philadelphia.

Halbwachs, N., Caspi, P., Raymond, P., and Pilaud, D. 1991. The synchronous data flow programming language LUSTRE. *Proceedings of the IEEE*, 79(9):1305–1319.

Harel, D. and Pnueli, A. 1985. On the development of reactive systems*Logics and models of concurrent systems*. Apt, K., editor. Springer-Verlag.

Ho, Y.-C. 1989. Dynamics of Discrete Event Systems. *Proceedings of the IEEE*, 77(1):3–6.

Johnson, C. 1992. Using temporal logic to support the specification of prototyping interactive control systems. *Int. Journal of Man-Machine Studies*, 37.

Lamport, L. 1977. Proving the correctness of multiprocess programs. *IEEE Trans. on Software Engineering*, 3(2):125–143.

Laplante, P. 1993. *Real-Time Systems Design and Analysis: An Engineer's Handbook*, IEEE Press.

Lawrence, P. 1989. Development of telerobotics for applications in natural environments. In *Workshop on Military Robotics*, Kingston, Ontario.

Lumelsky, V. 1991. On human performance in telerobotics. *IEEE Trans. on Systems, Man, and Cybernetics*, 21(5):971–982.

Manna, Z. and Wolper, P. 1984. Synthesis of Communicating Processes from Temporal Logic Specifications. *ACM Trans. on Programming Languages and Systems*, 6(1):68–93.

Murata, T. 1989. Petri nets: Properties, analysis, and applications. *Proceedings of the IEEE*, 77(4):541–580.

Ramadge, P. and Wonham, W. 1989. The control of discrete event systems. *Proceedings of the IEEE*, 77(1):81–98. Special Issue on Dynamics of Discrete Event Systems.

Reisig, W. 1992. Combining petri nets and other formal methods*Lecture Notes in Computer Science no. 616: Application and theory of Petri nets 1992*. Jensen, K., editor, pages 24–44. Springer Verlag.

Schilling, R. 1992. Telerobots in the nuclear industry. *Industrial Robot*, 19(2):11–14.

Sheridan, T. 1987. Telerobotics. In *10th IFAC World Congress on Automatic Control*, Munich, Germany.

Sheridan, T. 1992. *Telerobotics, automation, and human supervisory control*, MIT Press.

Sifakis, J. 1982. A unified approach for studying the properties of transition systems. *Theoretical Computer Science*, 18:227–258.

Spencer, H. 1992. Man in the loop. posting to the USENET newsgroup sci.space.

Stassen, H. 1990. Supervisory control behavior modelling: the challenge and necessity.*Robotics, Control and Society*. Moray, N., Ferrell, W., and Rouse, W., editors, pages 105–117. Taylor and Francis.

Thistle, J. and Wonham, W. 1986. Control problems in a temporal logic framework. *Int. J. Control*, 44(4):943–976.

Williams, O., Jackson, E., and Monteith, I. 1992. A control system for teleoperated and autonomous mining equipment exposed to sever moisture, corrosion and shock. In *5th Canadian Symp. on Mining Automation*, Vancouver, B.C.

Y03104

PROJECT JINGOSS[1]

A TELE-OPERATED METALLIC MINE DETECTOR (U)

Al Carruthers, Ron Eirich, Yoga Das
Defence Research Establishment Suffield
Box 4000, Medicine Hat, Alberta, Canada, T1A 8K6

Al Kramer
Oliphant Management Services
Box 36, Medicine Hat, Alberta, Canada, T1A 7G7

#149671

Abstract

On 14 January, 1993, the Canadian Forces Vice Chief of Defence Staff authorized the development of three remotely operated metallic mine detection systems, now nicknamed Jingoss. These were to be deployed with the Canadian contingent serving in Somalia. These units were to fulfil an urgent operational need for the detection and location of metallic anti-tank and anti-personnel land mines.

The Jingoss is a remote tele-operated vehicle which tows electromagnetic sensors. The sensors are optimized for the detection of metallic land mines. When a detection is made, the system automatically marks the ground and alerts the operator in the command vehicle which can be up to 5 km away. The electromagnetic signature of the object is displayed on the control station monitor, and the time and location of the detection are automatically recorded. A pan-and-tilt mounted video camera is mounted on the remote vehicle and the video signal is sent back to the operator's monitor. The operator controls the motion of the vehicle and the video camera with joystick controls.

Defence Research Establishment Suffield was tasked to develop and produce the units and ready them for deployment on a priority basis by 1 May, 1993. The systems were based on the integration of two technologies, the Vehicle Mounted Ordnance Detector and the Ancaeus Command and Control System, which were both under development at DRES. The two fields of ordnance detection and tele-operation of remote vehicles required accelerated development and integration to improve their performance for the mine detection role. The primary objective was to provide mine clearance personnel with a method of detecting mines which was faster and much safer than existing detection methods using hand-held mine detectors. The resulting Jingoss system provides a reliable system of detection which greatly improves the safety and productivity of personnel employed in this hazardous work of clearing mines during peacekeeping operations.

Introduction

The problem posed by land mines is of considerable concern to Canadian Army units. More recently, this threat has been experienced during peacekeeping operations, often with disastrous results. In order to at least partially counter this threat, Defence Research Establishment Suffield (DRES) was tasked by National Defence Headquarters (NDHQ) to develop and manufacture three tele-operated systems for the detection of metallic land mines. This requirement was authorized in January 1993 by the Vice Chief of Defence Staff as an urgent operational need for the Canadian contingent deployed in Somalia. The primary criteria for the remotely operated metallic mine detection system, nick-named "Jingoss" , was to improve the overall safety for personnel working or travelling in areas suspected of having emplaced land mines or boobytraps. Other significant criteria were that military engineer personnel were to be trained to operate and maintain the system and that the systems be ready for deployment to Somalia no later than 1 May, 1993. This paper describes the development, operating characteristics, and introduction into service of the Jingoss system. Particular emphasis has been given to the operational usage or application of a tele-operated system in a military peacekeeping scenario.

---

[1] "Jingoss" is an Algonquin word describing "the actions of a ferret".

Development and Concept of Employment

Over the years, close communication has developed between the military engineer requirements directorate, Director Military Engineering 4, and DRES. This working relationship enabled the rapid exchange of information and technical details during the feasibility phase of the project. DRES had been working in two technological areas which could be directly applied to the mine detection application. These areas were the vehicle mounted ordnance detector (VMOD) and a semi-autonomous ground vehicle control system (Ancaeus).

It quickly became apparent that if the assigned timetable were to be met, maximum use had to be made of the existing VMOD and Ancaeus technologies. Development had to be limited to adapting the existing systems to the mine detection role and integrating them into a reliable system which could be fielded in a peacekeeping scenario. To minimize support requirements in Somalia, the basic chassis for the Jingoss system was to be an eight-wheeled all-terrain vehicle called the ARGO. This vehicle was already in the Canadian Forces inventory and its simple, robust construction made it desirable from the users point of view. The command vehicle likely to be used was an eight-wheeled armoured personnel carrier known as the Bison. This vehicle would operate as the base station during remote operation of the Jingoss system and as the carrier for the combat engineer section. A combat engineer section of eight men were to operate and maintain the Jingoss system and to provide an immediate response capability to deal with any detections that were made.

The concept of employment that evolved during the feasibility phase of the project was that a tele-operated vehicle would tow a metallic mine detector array. The remote vehicle would be controlled by an operator located in a command vehicle which remained at a safe distance from the remote vehicle in the event of any detonation of mines or other ordnance. The operator is warned in real time of any detections made by the unit, and the location is automatically recorded and marked by the system. The commander of the engineer section would then determine the best method of investigating the detection and take any necessary action to neutralize, dispose of, or bypass any hazardous ordnance which may be present. A summary of the required operating characteristics of the system is given in Table I.

Table I. Operating Characteristics

PARAMETER	VALUE
Scan Width	4 m (reducible to 2m)
Scan Speed, typical	5 km/hr
Detection Capability	
-Large metallic AT mine	>98% to 0.7 m depth
-Large metallic AP mine	>95% to 0.3 m depth
-Fragmentation Grenade	>95% to 0.1 m depth
Marking Accuracy	2 m radius
Operating Day	up to 12 hours
Remote Operation Range	up to 5 km
Terrain	roads or smooth terrain
Crew Size	2 persons

System Overview

A photograph of the complete Jingoss system in operation is shown in Fig. 1. The system consists of three major components: the remote controlled vehicle (RCV); the command vehicle carrying the Ground Control Station (GCS) and Jingoss operator; and the transport trailer described below:

Remote Controlled Vehicle

The tow vehicle is a commercial skid-steered vehicle manufactured by Ontario Drive and Gear, Ltd., which had to be substantially modified for integration into the Jingoss system. The vehicle is used to tow the mine detector sensor platforms and carries all the electronic subsystems of the detector as well as those of the Ancaeus control system. Mounted in front of vehicle is a "chain sweep" that drags a heavy steel chain over the ground to disrupt or detonate small anti-personnel mines. The overall weight of the vehicle and point loading has been kept well below the activation pressure required to detonate pressure fuses

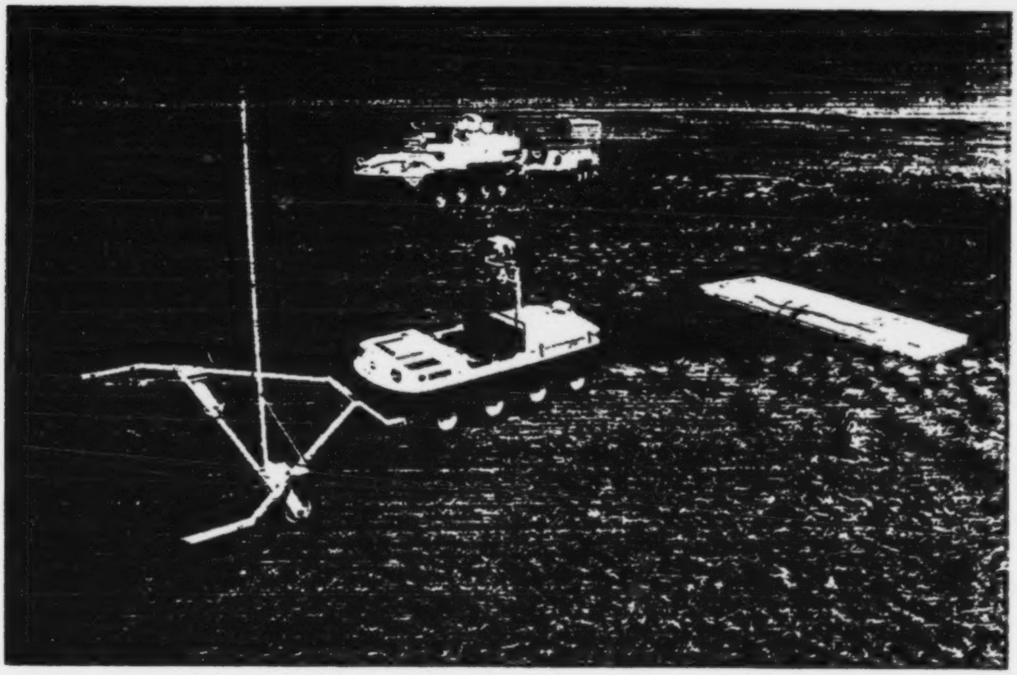


Figure 1. Complete JINGOSS system.

of anti-tank mines. The tall mast on the chain sweep is used to engage overhead trip wires before the command vehicle encounters them.

Command Vehicle

The ground control station is mounted inside a Bison APC. The operator observes the monitor and controls the movement of the remote vehicle with a custom joystick control. In addition he controls the functions of the mine detection and marking systems through a sophisticated user interface. The operator is presented with all relevant status and performance information for the remote vehicle and the detection system.

Trailer

A specialized trailer is provided for each system which will transport the entire system and all the spare parts and supplies required for operation and maintenance. The remote vehicle is stowed in the main compartment of the trailer and the sensor platforms are stowed beneath the main compartment floor. The chain sweep disassembles and stores in the main cargo area. Other smaller components, spares, fuel, tool kits, etc. are stored on shelves or in compartments. In essence, the Bison and trailer form a self-contained system which can be easily transported. The Jingoss system can be quickly deployed by road to a given location or airlifted by C130 transport aircraft.

The Detector System

The detector system was already in an advanced stage of development in the form of the VMOD. More detail on the VMOD system are given in Refs [1] and [2]. Ref [3] provides a detailed description of the changes and the rationale for those changes to the VMOD system to prepare it for the mine detection role. This paper summarizes the detection and marking system features and characteristics which were incorporated into the Jingoss system to optimize its detection performance against metallic land mines. The modifications were made to the VMOD system by Pylon Electronics Limited of Ottawa, Ontario. Pylon also produced a total of three detection units and spares for the Jingoss systems.

It had to be accepted from the outset that there was no known practical technology available in the world which would reliably detect non-metallic mines. The nature of the mine threat in Somalia was known to be primarily older generation mines and therefore had a significant metal content. This factor made the use of a metallic detector a practical system for use in the Somalia peacekeeping scenario.

A primary requirement was provision of a scan width wide enough to be practical for scanning roads or other wide surfaces but still retain the required capacity and reliability for detection of the threat mines. Feasibility tests were conducted at DRES to verify whether or not a modified VMOD could detect mines at the rate and detection probabilities required. This required testing of new sensor coil design and configurations. Another issue that had to be addressed was the discrepancy between the data rate used on the VMOD (19.2 kbaud) and the data rate used by the Ancaeus control system (9600 baud). Estimates were made and confirmatory testing was done to ensure that sufficient detector sensitivity and data rate transfer could be achieved by using the lower 9600 baud rate. Once these issues were resolved, Pylon was contracted to produce the revised detector systems. Some important operational features retained by the Jingoss system which were available on the original VMOD include:

a. Indication that the detector system is operating properly.
b. Audio and visual indication of detection.
c. Running display of two channels of response vs time data (compressed to be compatible with Ancaeus baud rate).
d. Ability to change system sensitivity (coarse selections only, i.e. low, medium and high).
e. Ability to turn lane markers and detection markers on/off and to adjust the interval between lane markings.
f. Ability to adjust threshold detection levels based on ambient conditions.
g. Indication of hitch disconnect.
h. Ability to start/stop system operation with ease.
i. Ability to recalibrate/test detector system easily.

The VMOD operates at 50 Hz and appropriate digital filtering is used to minimize interference due to 60 Hz power lines. The Jingoss was to be deployed in an area using 50 Hz power, and therefore the operating frequency was changed to 70 Hz and appropriate digital filter parameters were applied. This allows the system to operate in both 50 and 60 Hz environments. As well, since the detectors work directly in the RF environment of the Ancaeus system transmitters, great care was taken to provide proper grounding and shielding to minimize interference. The ARGO vehicle itself was not designed with stringent electrical noise suppression in mind. In addition, the ARGO engine is in close proximity to the sensor platform during operation. During modifications to the ARGO vehicle, considerable effort was devoted to reduce the noise levels generated by the ARGO to an acceptable limit.

The VMOD utilized a marking system which deployed a foam as a marking agent. The foam-based system was known to behave poorly in a very hot climates and a substitute marking agent had to be identified. Several alternatives were considered such as chalk, water-based paint, diesel-based aluminum paint, and dyes, however, a marking system using only diesel fuel was selected. This was based on the transient nature of the fuel which would evaporate within days and the availability of the diesel in Somalia. A marking system using pressurized diesel with no additives was designed and built by Pylon for integration into the Jingoss system.

The Ancaeus Control System

The concept and preliminary development of the Ancaeus control system were described at the Third Conference on Military Robotic Applications in September 1991 [Ref. 3]. That presentation described the Ancaeus concept which is a generic control system capable of guiding practically any surface vehicle and navigating semi-autonomously to deliver an application payload. The Ancaeus electronics package which was utilized in the Jingoss project is outlined in Figure 2. The package is modular in nature which facilitates rapid repair of the system in the field and provides a wide scope for system flexibility and adaptation. These features made the Ancaeus control system ideally suited for integrating the tele-operated vehicle and the mine detection system into a Jingoss system package.

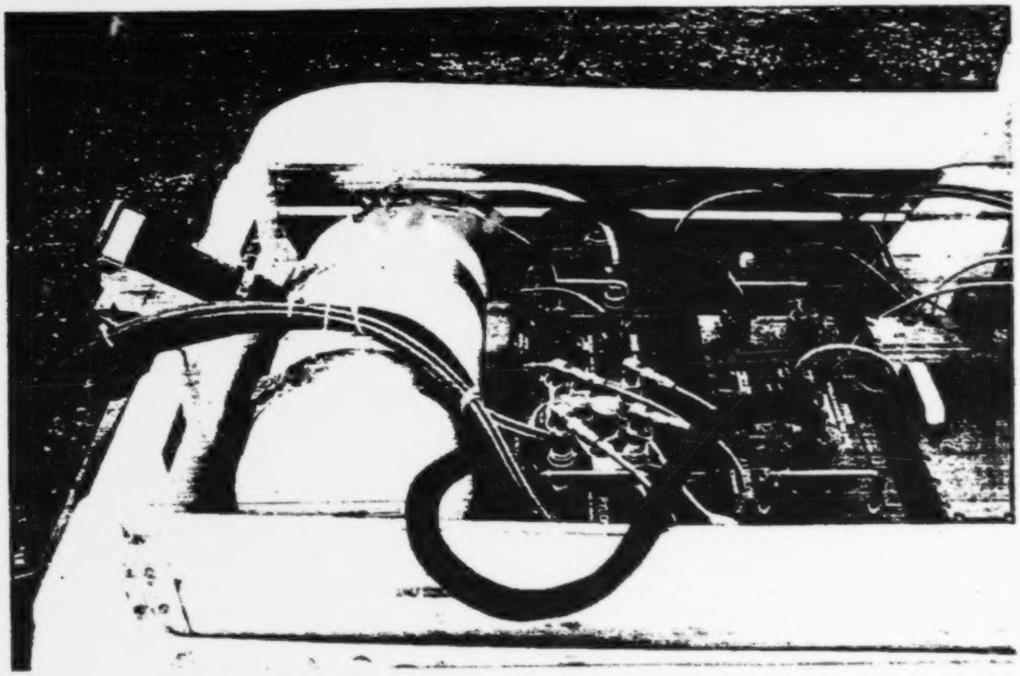


Figure 2. Remote vehicle electronics compartment

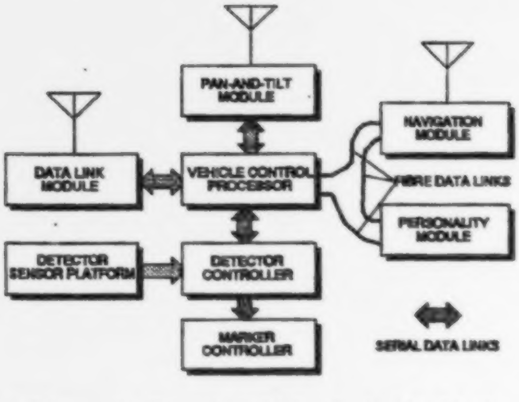

PAN-AND-TILT MODULE
NAVIGATION MODULE
DATA LINK MODULE
VEHICLE CONTROL PROCESSOR
FIBRE DATA LINKS
PERSONALITY MODULE
DETECTOR SENSOR PLATFORM
DETECTOR CONTROLLER
MARKER CONTROLLER
SERIAL DATA LINKS

Figure 3. Remote vehicle electronics block diagram.

The Ancaeus remote vehicle hardware consists of five major modules plus the application modules, i.e., the mine detection and marking systems which were described above. The majority of the electronics package for the vehicle is carried in the rear cargo compartment of the ARGO. (Figure 2). This compartment was covered with a plastic shield to protect the contents from the direct rays of the sun and inclement weather. Interconnection between the various modules is by cables or fibre optic links when high data volumes must be passed. Twenty-four volt DC power is supplied to each module from storage batteries which are charged by a separate alternator on the ARGO vehicle engine.

Vehicle Control Processor Module

The vehicle control processor module functions as a network gateway between the control data link and the fibre optic data link. Higher level functions of the vehicle such as semi-autonomous navigation also reside in this module. The module is housed in a rugged, temperature controlled, weather proof enclosure. Temperature extremes are controlled by the use of a thermo-electric heat pump which provides heating or cooling as required. This controlled environment allowed the use of commercial grade components on the remote vehicle despite planned deployment to extremely hot or cold climates. DC to DC converters convert the 24 volt DC vehicle power to the required voltages for the on-board processors. The software for the vehicle control processor is written in "C" and speed critical routines have been written in assembler. The code resides in Erasable Programable Read Only Memory (EPROM).

Vehicle Personality Module

The Ancaeus concept makes maximum use of common modules in all its potential applications. A minimum number of changes to specific modules is all that is reqired to make an operating command and control system. The vehicle personality module was specifically designed for remotely controlling the ARGO vehicle. In the process it was "ruggedized" to endure the harsh environment of being mounted in the engine compartment of the vehicle and greatly simplified from previous versions to enhance reliability. The personality module controls all vehicle functions such as starting and stopping the engine, steering, cruise control, and braking. It also monitors the status of the vehicle by transmitting status reports on items such as engine temperature, charging current and voltage, vehicle speed, and other useful vehicle parameters. The cruise control feature was most useful in the Jingoss application because the operator could set an optimum search speed for the vehicle which would be accurately maintained without operator input. This allowed the operator to concentrate on the mine detection aspects rather than having to regulate vehicle speed. In actual operation, the driving consists mainly of simply guiding the vehicle down roads or across relatively smooth terrain.

Navigation Module

All functions associated with determining the position of the vehicle reside in the navigation module. The module uses a global positioning system (GPS) receiver to obtain navigational data from orbiting satellites. The GPS location data is accurate to a distance of approximately 75 metres and can be displayed in longitude/latitude or military grid reference format. This accuracy could be improved upon using differential GPS techniques; however, the main purpose of the GPS was to assist in the reporting of hazardous areas. The actual location of each mine would be marked as described above. The navigation module also contains a tilt and a roll sensor to alert the operator of a dangerous vehicle condition.

Data Link Module

The data link module contains the RF modem and RF amplifier. The data link communicates all guidance commands, status reports and detection data between the remote vehicle and the command vehicle. Sufficient power was provided in the Jingoss system for data and video transmission at distances up to 5 km. During the familiarization training and the development of clearance procedures it was found that a distance of approximately 200 metres was optimal. This allowed visual contact with the remote vehicle most of the time and allowed the command vehicle and personnel to remain outside the lethal radius of any ordnance that might be detonated.

Pan-and-Tilt Module

A pan-and-tilt unit fitted with a CCD colour video camera is the "eyes" of Jingoss. The upper head of the module will rapidly turn left and right 180 degrees to provide all-round vision for the operator. The head can be tilted up or down to view objects above or below the horizontal plane. The camera has a fixed wide angle lens and an aperture control which can be manually adjusted for varying light conditions. Two banks of flood and spot lights are supplied for night operation and supplement the vehicle headlights. The base of the pan-and-tilt module houses the video transmitter, cooling fans, and supports the video antenna. Immediately below the pan-and-tilt unit is a warning strobe light to indicate when the vehicle is operating under remote control.

Command Vehicle and Control Station

Virtually any vehicle can be used for the command vehicle. The only modifications required are: the provision of an inverter to change vehicle power to 110 volts AC; mounts for the video and data link antennas; space to mount the control station; and, room for the Jingoss operator. If the ambient temperature inside the vehicle is likely to exceed 40 degrees Celsius, air conditioning should be provided to keep the electronic equipment cool. The air conditioning would no doubt be appreciated by the personnel operating the system as well. The Canadian Forces have selected an eight-wheeled armoured personnel carrier known as the Bison for the command vehicle. This is due to the inherent protection offered by the vehicle's armour and the fact that it can transport a

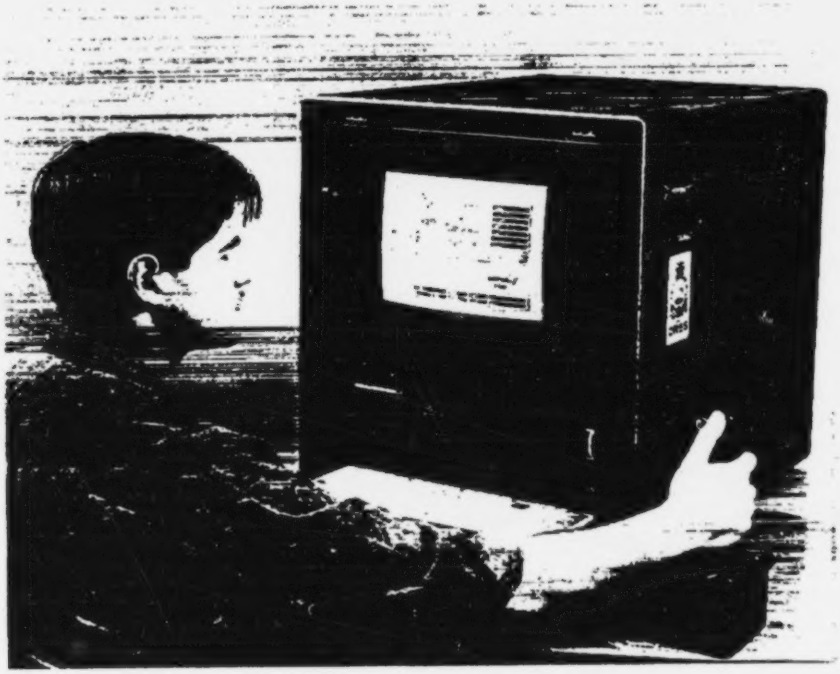


Figure 4. Control station

field engineer section together with the Jingoss system. The engineer section will provide the operators for the Jingoss as well as providing an immediate capability to investigate detections and deal with them as necessary.

The control station (Fig. 4) is shock-mounted in a durable weather resistant cabinet within the command vehicle. The components within the cabinet are modular and assembly is largely a matter of plugging in the appropriate module. The modular approach was taken to facilitate maintenance by the field engineers operators. All communication between the tele-operated vehicle and Jingoss operator takes place via the control station through a RF modem and a video receiver. The computer used has been configured with "ruggedized" components including the monitor, hard drive and keyboard. It has been equipped with the necessary video insertion hardware to provide a relatively sophisticated display for the operator. The operator observes the monitor which shows whatever displays he selects. He selects the displays by means of pull-down menus, keyboard inputs, or by pointing a cursor and selecting the appropriate icon on the computer screen. The vehicle steering, braking and acceleration are controlled by means of a sophisticated joystick. On the handle of the joystick control is a thumb-operated mini-joystick which controls the movement of the camera pan-and-tilt unit on the ARGO vehicle. An emergency stop button is located on the joystick handle and right and left mouse buttons are replicated by another button on the joystick handle and a joystick trigger respectively. Cursor movement on the monitor is done using a standard computer mouse or can be done using the joystick.

When the field engineers were being trained on the operation and maintenance of the Jingoss system, it was apparent that the operation of the vehicle was not going to present any problems. This was primarily due to the effort which was put into the "C" software to make it "user friendly" and convenient for the operator. With a minimum of practice, personnel became quite adept at the control of the vehicle and interpreting the data which was sent back from the detection system. The number of displays or data that the operator views on the monitor is at his discretion and can be arranged on the screen according to his personal preference.

Among the displays available to the operator are:

a. The view from the video camera mounted on the ARGO vehicle for remote operation of the vehicle. The camera can be rotated or tilted in whatever direction the operator desires.

b. A video-audio window which permits the operator to turn on and off the camera, spot and floodlights, and the audio microphone on the pan-and-tilt unit.

c. A vehicle system status window which provides the operator with information such as engine temperature, vehicle speed, transmission gear selected, engine RPM, fuel level, battery voltage and current, and vehicle pitch and roll. The real time location and heading of the vehicle are also displayed and continuously updated.

d. The mine detector control panel gives the operator complete control over the detection and marking systems mounted on the remote vehicle. He can turn the right and left lane markers on/off and adjust the interval between the marks. He can manually mark points on the ground in addition to the automatic marking of detections made by the system. The mine detector is calibrated during the initial start-up of the system and remotely recalibrated periodically to adjust for the background level of magnetic signal in a particular area. The recalibration takes approximately 15 seconds to perform. Sensitivity of the detectors can be adjusted to a high, medium or low level. The display will also indicate if the platform has separated from the remote vehicle when a solid obstacle triggers the release of the hitch to prevent damage to the detector platforms.

e. A graphic display is available for any detection received by the unit. Detections on the left or right coils are differentiated by different colours for the profile lines shown on the display. The slope and amplitude of the line displayed gives the operator a rough indication of the nature of the object detected.

f. A history panel can be called up to display a record of the detections and their location. This can also be printed out as a permanent record by the on-board printer.

g. The operator can call up a moving map display for the operational area that the system is operating within. The actual position of the vehicle and the vehicle heading is overlaid on the map display. The map data must be scanned onto floppy disks and loaded into the control station computer via a port on the front of the computer.

Introduction Into Service

A great number of factors had to be considered during the equipment development and production to permit its deployment to a harsh environment half-way around the world. Furthermore, the situation in Somalia would not permit sending DRES support personnel during the initial fielding of the equipment. Throughout the design and development, components were specified to the highest practical level for robustness and reliability. Some components were simply not available and additional protection was provided. A good example of this was the air conditioned enclosure provided for more fragile components on the remote vehicle.

The remote theatre of operations for the Jingoss system meant that many other precautions had to be taken to ensure equipment serviceability. The modular nature of the system allowed easy replacement of defective components in the field. Therefore a higher than normal scale of spare parts were to be provided. Defective components would have to be sent back to Canada for diagnosis and repair because the support agencies in Somalia were not trained or equipped to maintain most of the Jingoss components. As a general rule the crew of field engineers would be responsible for replacing components or sealed "black boxes" in the field and ensuring that an adequate stock of spares was maintained. To assist in the re-ordering of spares and repair parts an illustrated spare parts list was issued to the crew deployed with the equipment. Provision of this essential item was a real challenge because of the short time available and the many modifications being made to the system right up to the date of deployment. Similarly, operator manuals and aide memoirs were being written at the last minute to incorporate the knowledge and experience gained from the training of the CF engineer section who would be going to Somalia with the equipment.

The enthusiasm and professionalism of the field engineer personnel being trained on the Jingoss system was most appreciated by the people at DRES. They displayed a keen willingness to learn and a cooperative attitude which enabled the necessary knowledge and unfamiliar material to be digested in a minimum of time. The engineer trade is traditionally employed on tasks such as demolition and bridge building, and therefore much of the remotely operated hardware and electronic equipment was very new to them. In spite of this, their collective skills and experience were applied to the training and they became proficient operators and skilled at maintenance and repair of the unit in a very short time. In addition to this, the field engineer section of eight men developed and verified detailed operating procedures for the use of the equipment in an operational scenario. They produced a comprehensive aide memoir which included routine maintenance and servicing, daily inspections, equipment assembly, and troubleshooting charts for on-scene diagnosis and repair. Detailed procedures were developed for the handling of mines and other hazardous items which were detected by the system. During the practical exercises conducted as part of their training, data was collected on detection rates and how to maximize the productivity of the crew without jeopardizing personnel safety. Much of the enthusiasm displayed by the field engineers was the result of confidence gained in the reliability of detection of all of the dummy mines utilized during the training and the speed relative to existing methods using hand-held mine detectors.

Conclusion

There were many observations and conclusions derived from the Jingoss Project. Foremost was the fact that an operational need can generate the required impetus to field in a very short time a practical tele-operated system based on currently available technology. The existence of the VMOD and Ancaeus technologies in an advanced state of development greatly reduced the technical risk and the equipment design was sufficiently flexible to permit modification for use in the metallic mine detection application.

The Ancaeus system with its modular construction and inherent versatility was a key element in the rapid integration of the current overall system. Those characteristics permitted concurrent development of the many system modules which were then integrated rapidly as the project neared completion. The powerful capability of the Ancaeus system was never overloaded by the mine detection application and it exhibited considerable growth potential for add-on capability to the Jingoss system or use in other applications. Other applications being pursued at DRES include a hazardous material handling robotic vehicle, mobile target systems, and tele-operated anti-tank mines. The Jingoss equipment itself showed a potential for other uses such as a robotic reconnaissance vehicle and a detection system for unexploded ordnance in range or old battlefield clearance tasks. It is clear that simplistic tele-operated robotic systems are well within the capability of today's technology and can provide a valuable service in numerous military scenarios.

References

1. Das, Y., Toews, J.D., and McFee, J.E., "Vehicle Mounted Ordnance Locator: An Experimental Prototype (U)", Suffield Report No. 434, DRES, Ralston, Alberta, Canada, February 1988.

2. Das, Y., McFee, J.E., and Toews, J.D., "Development of a Vehicle Mounted Ordnance Detector (U)", Proceedings of the ABCA-5 Meeting, Toronto, Ontario, Canada, June 4-6, 1991.

3. Das, Y., Carruthers, A.R., Toews, J.D., McFee, J.E., and Elliott, J.R., "Remotely-Operated Metallic Mine Detector System (ROMMIDS) (U)", Presentation to the ABCA-5 Meeting, HMAS WATSON, Watsons Bay, Sydney, Australia, October 11-15, 1993.

4. Eirich, R., and Kramer, A., "A Generic Semi-Autonomous Ground Vehicle Control System", Proceedings of the 3rd Conference on Military Robotics Applications, Medicine Hat, Alberta, Canada, September 9-12, 1991.

414959

Paradigms of Intelligent Robotics

Jagdeep S. Basran[1] and Emil Petriu
Department of Electrical Engineering
University of Ottawa
Ottawa, Ontario
K1N 6N5

#149673

Abstract

Using the framework of paradigms, where a paradigm refers to a group of researchers who behave as a community, this paper presents a broad perspective on intelligent robotics. This perspective includes various items such as terminology, history, and current paradigms. The meanings of the terms intelligence, robotic system, intelligent robotic system, and associated terms are examined and related to the paradigm concept. The relationship between artificial intelligence and robotics, both historical and ongoing, is discussed. The current paradigms of intelligent robotics are connected to the emergence of several disciplines in the last half century. These include the viewpoints of the control, artificial intelligence, robotic, and modelling and simulation communities, which are briefly described.

Introduction

A paradigm refers to a community of researchers who share a common set of ideas and conventions regarding a particular subject matter. The conventions of a paradigm are established over time as a group of researchers implicitly or explicitly accept certain conventions, while disregarding or rejecting others. At any specific time, the existing paradigms provide both a substrate for change and strong resistance to change. Paradigms are dynamic entities that have been recently established, are going strong, or are slowly fading, as various researchers continue to form groups and interact socially. The formation of new paradigms usually involves a critical assessment of existing conventions and may involve bringing together ideas from a variety of sources. The current publications of the scientific world reflect the presently accepted communities of researchers. This concept of paradigms as a means to view the scientific enterprise including the nature of scientific progress was introduced in [Kuhn, 1962]. Kuhn's original analysis, which was developed for the physical sciences, was adapted and extended in its application to the field of psychology [Lachman et al., 1979]. More recently in [Tapscott and Caston, 1993], the concept of paradigms and paradigm shifts are used to discuss how the changes in information technology should affect the way that business is carried out. In this paper, the notion of a paradigm involves a group of individuals who are doing related research, reference each others' work, and most likely evaluate each others' work too.

During the last half century, a number of fields have emerged, including control theory, digital computing, operations research, information theory, cybernetics, system theory, artificial intelligence, and modelling and simulation [Klir, 1985]. Each of these fields has a history of development which is in many ways interconnected with the history of related fields. These histories established the dominant conventions associated with each of these fields. In addition, over time, each of these fields have produced many different paradigms. That is, most fields are multiparadigmatic.

The field of robotics is a place where ideas can meet realities. This meeting provides an opportunity for evaluating the ideas. The ideas that are generated are a reflection of the realities and the realities that are constructed are molded by the ideas. Ideas and realities mutually shape one another and are interconnected with each other [Schultz and Lavenda, 1990]. As such an outstanding meeting-place, the field of robotics is where researchers from a multitude of paradigms can bring their ideas and realities together. This multiparadigmatic quality of the field of robotics is even further amplified by the inherently multidisciplinary nature of this field.

The purpose of this paper is to explore how one facet, the facet of intelligent robotics, is seen from the viewpoints of different research communities. Before presenting these viewpoints, we must address two other issues. First, we examine certain terms and their meanings. Second, we explore the relationship between artificial intelligence and robotics.

Terms

The use of words in a language, in this case English, occurs in a context. The meaning of a term varies considerably with the context of use. The paradigm of a researcher provides the context with which meaning is given to the terms in use [Lachman et al., 1979]. This link between semantics and paradigms is important. This section examines the terms: intelligence, robotic system, and intelligent robotic system.

---

[1] Any correspondence regarding this paper can also be e-mailed to 'basran@iit.nrc.ca'.

Intelligence

In most cases, humans are used to calibrate the meaning of intelligence. Human intelligence has many facets and the purpose of research into intelligence is usually to explore one or more of these facets. However, the goals of a researcher significantly affect how she or he approaches the exploration of intelligence. The goal of obtaining a better understanding of certain aspects of human intelligence through fields such as neurophysiology, psychology, or cognitive science is approached from a different viewpoint than the goal of implementing certain aspects of human intelligence on a machine.

One of the first discussions on the relationship between intelligence and machines occurred in the seminal papers of Alan Turing. He proposed a specific test as a means to determine whether a machine had successfully achieved some level of intelligence. The details of this test are not important here. What is important is that the Turing test was designed in such a way that the intellectual aspects of machine intelligence were evaluated, while the physical aspects of this test were minimized [Turing, 1950]. Turing recognized the approach of building a physical replica of a human as a means of developing machine intelligence but at the time this approach seemed "too slow and impracticable" [Turing, 1948]. Another interesting perspective was the one that views intelligence as an n-dimensional space where animals, humans, and machines occupy different regions of this continuum [Armer, 1960]. Some of the open questions are what the possible dimensions of this space are and where animals, humans, and various machines are located in this space.

More recently, a theory of intelligence that encompasses both natural and artificial systems has been outlined [Albus, 1991]. Albus defines intelligence as that which produces successful behavior and he identifies the elements of intelligence as sensory processing, world modeling, behavior generation, and value judgement. This paper places an emphasis not on the actual elements of intelligence but on the organization and communication, or architecture, of these elements in an intelligent system. The proposed architecture is basically an extension of the NASREM (NASA/NBS[§] (National Aeronautics and Space Administration/National Bureau of Standards) Standard Reference Model) architecture [Albus et al., 1987] to encompass living systems, including humans. Another investigation of general intelligence involves the work of [Newell and Simon, 1972; Rosenbloom et al., 1991]. The primary focus of this work is on building a computer simulation of human problem solving where symbol processing is the key to intelligence [Norman, 1991]. The nature of intelligence and the approach to its development is viewed quite differently in the research of [Brooks, 1991a, 1991b]. Brooks proposes an incremental approach to the development of intelligence that is based upon dynamics of interaction with the world. This approach has been explored with indoor mobile robots. He advocates a physical test for machine intelligence. He also rejects the importance of representation and reason for intelligence, as understood by the symbol processing approach. For a more complete discussion of these issues refer to [Kirsh, 1991a, 1991b].

Robotic System

We use the term, robotic system, as a general term that encompasses the full range of systems that researchers view as robotic. This subsection examines this range of robotic systems and makes connections between the terms used to describe particular robotic systems and different paradigms of robotics.

The field of robotics is multiparadigmatic and three key factors help distinguish between these paradigms. The type of robotic system is one crucial factor. The type of robotic system is determined for the most part by its physical construction and operational characteristics, which are usually interdependent. A key operational attribute is the degree of autonomy of a robotic system which can vary along a continuum from teleoperated to autonomous. A second factor that distinguishes robotic paradigms is the particular subject matters emphasized by the researcher. Is the focus on sensing, control, programming, planning, navigation, manipulation, learning, intelligence, architectures, or some combination? Lastly, the environment or environments that a robotic system can function in are used to make paradigmatic distinctions [Woods, 1989; Desrochers, 1992]. The dependence between the environment and the type of robotic system, that is constructed and used, is an important feature of the nature of these paradigmatic distinctions.

To clarify these ideas, we apply this framework to certain, existing robotic systems. In the context of a manufacturing environment, an industrial robot is a robotic system which has a rigid construction often like an arm, can manipulate its environment with some level of skill, operates autonomously, and has limited mobility. An automated guided vehicle (AGV) is a robotic system which is rigid, might have a manipulator, is often involved in material handling tasks, and has mobility limited usually to the guidepath.

[§] Presently called NIST, the National Institute for Standards and Technology.

A discussion of the different paradigms of mobile robotics with an emphasis on language further clarifies these ideas. To begin with, an AGV is a mobile robot that is constrained to follow a guidepath on the factory floor. The term, mobile robot, usually refers to a robotic system which has no manipulators and whose means of locomotion is power-driven wheels, that limit operation to indoor or relatively flat outdoor environments. The term, legged robot, refers to a robotic system whose means of locomotion are legs which physically might be adapted manipulators and whose intended environment of operation is usually more rugged terrain. The term, vehicle, is being currently used for land vehicles whose goal is to navigate within the human transportation system of freeways, highways, roads, and streets. Although not exclusively, the term, rover, applies to a mobile robotic system which operates on the surface of planets other than Earth. The degree of autonomy for mobile robotic systems varies from teleoperated to autonomous. The paradigms of mobile robotics typically emphasize the issue of navigation within a particular environment rather than manipulation.

Intelligent Robotic System

The combination of, varieties in the meaning of intelligence and the types of robotic systems, makes the question of what defines an intelligent robotic system particularly challenging; however, a number of authors have discussed this topic [Birk and Kelley, 1981; Gevarter, 1982, 1983, 1985; Miller, 1983; Hunt, 1985; Nitzan, 1985; Brady, 1985; Inoue, 1985; Brooks, 1986b; Staugaard, 1987; Lee, 1989; Tzafestas, 1991]. Regardless of the different categories used to describe the various generations of robotic systems in these references, the final generation is always the robotic system that is intelligent. Although the ultimate goal may be the same, what defines such systems does vary.

To paraphrase the well-known statement in [Brady, 1985], an intelligent robotic system is that system which intelligently connects perception to action for a particular paradigm. Figure 1 illustrates this simple framework for intelligent robotic systems. Several questions arise out of this statement. What form must an intelligent connection between perception and action take? For a number of researchers, an intelligent connection involves applying the techniques of traditional artificial intelligence (AI), which emphasizes certain aspects of cognitive processing such as modelling, planning, and reasoning. What should an intelligent robotic system be able to perceive? According to [Nitzan, 1985], an intelligent robot should be able to perform a variety of tasks under conditions not known a priori. This implies the ability to perceive unpredictable conditions (including faulty operation of self), plan what to do under such contingencies, and then execute the developed plan. What actions if executed by a robotic system do researchers consider intelligent? As was previously mentioned, manipulating physical objects is a key intelligent action for most industrial robots, while navigating in a specific environment is a key intelligent action for most mobile robots. How does the paradigm of a researcher affect the answers to the previous questions? The problem of making industrial robots operate intelligently in a manufacturing environment has been the primary focus of [Miller 1983; Lee 1989]. The simple responses to these questions were given for exemplary purposes and are examined in greater detail later.

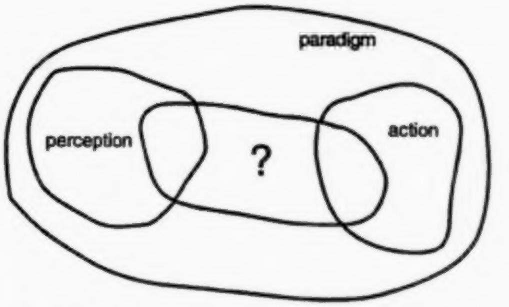


Figure 1: A simple framework for intelligent robotic systems

The adjective, intelligent, is an overused word that has been applied to several components of and tasks executed by an intelligent robotic system. This adjective has been applied to sensing, control, planning, navigation, assembly, welding, and so on. To what degree developments in an intelligent robotic system depend on these intelligent subsystems is an interesting question. Discussions on intelligent robotics often lead to how the various techniques of AI can be applied to robotic systems and thus, endow these systems with intelligence. This relationship between the fields of robotics and AI is an important one and will be explored in the next section.

An Historical Perspective

Is the field of robotics a subfield of AI concerned with perceptual and motor tasks, or is the field of AI a subfield of robotics concerned with cognition [Rich and Knight, 1991]? Maybe the field of robotics and the field of AI both have multiple paradigms, whose borders are ambiguous and sometimes overlap when there is mingling between paradigms. The relationship between these two fields has been explored by a number of authors [Gevarter, 1982, 1983, 1985; Miller, 1983; Hunt, 1985; Brady, 1985; Mowforth and Bratko, 1986; Staugaard, 1987; Lee, 1989]. Generally, the presentation in these references is to take a snapshot of AI, a snapshot of robotics, and then to explain how the existing AI can or has produced intelligent robots.

Table I: Historical highlights of robotics and artificial intelligence

Industrial Robotics		Research-oriented Robotics		Artificial Intelligence
	1948	teleoperation with force feedback	1948	*Intelligent Machinery*
	•		•	
automated guided vehicles	1956	micromanipulator	1956	Dartmouth conference
gantry robot	1957		1957	general problem solver
first industrial robot arm	1958	teleoperation of radioactive	1958	developed LISP (list processor)
	1959	material and for undersea activity	1959	geometry theorem proving
	1960		1960	
diecasting with robots	1961		1961	symbolic integration
	1962		1962	
	1963		1963	*Computers and Thought*
spot welding robots	1964		1964	chemical analysis expert system
spray painting robots	1965	prosthetic hand	1965	simple computer vision
	1966		1966	natural language understanding
	1967	walking machines	1967	
industrial robots built in Japan	1968	SHAKEY robot	1968	SHAKEY robot
	1969		1969	
Symposium on Industrial Robots	1970	teleoperation of Russian rovers	1970	*Artificial Intelligence periodical*
Japanese Industrial Robot	1971	on surface of moon	1971	robot planning program
Association	1972		1972	understanding spoken language
electric drive robot	1973		1973	
	1974		1974	
integrated computer aided	1975		1975	
manufacturing project	1976	United States Viking landers	1976	infectious diseases expert system
	1977	on Mars	1977	geologist assistant expert system
VAL robot program released	1978	voice-controlled wheelchair	1978	
machine vision in industry	1979		1979	naive physics understanding
	1980		1980	commercial expert systems
weld seam tracking	1981	Canadian robot arm in space	1981	*Handbook of AI*
	1982	*Journal of Robotics Research*	1982	Japanese 5th generation project
tactile sensors	1983	autonomous land vehicle projects	1983	
Robotics and Computer Integrated	1984		1984	United States Cyc project
Manufacturing	1985	*Robotics and Expert Systems*	1985	qualitative reasoning
Robotics and Automation	1986	subsumption architecture	1986	
Conference	1987		1987	
	1988	*Journal of Intelligent and*	1988	
	1989	*Robotic Systems*	1989	*Principles of Knowledge*
	1990		1990	*Representation and Reasoning*
	1991		1991	*Conference*
	1992		1992	
	1993		1993	

A glimpse of the history of robotics, the history of AI, and some connections between these two are presented with the aid of Table I[†]. Several comments regarding the nature of this table are necessary. First, this table shows two main streams of modern robotics [Saveriano, 1988]. The industrial robotics stream is concerned with the effective use of industrial robots and AGVs within the current framework of computer integrated manufacturing. Meanwhile, the research-oriented robotics stream is much more open-ended than the manufacturing stream and has evolved from a diversity of different strands from its beginnings. Second, the history of AI that is displayed represents the dominant symbol processing approach of traditional AI. The history of neural networks, which are categorized as simply one type of learning by traditional AI [Rich and Knight, 1991], is not included[‡]. Lastly, the highlights of history that are shown in the table are obviously very selective and events from histories of related fields such as computing, cognitive psychology, or cybernetics are not included.

The mobile robot SHAKEY, developed at the Stanford Research Institute International in the United States, is mentioned in the columns of both AI and research-oriented robotics, since this project represents the first significant attempt to investigate the common research issues of these fields in an actual physical environment. This project was described as an application of AI techniques in the development of an integrated robot system [Nilsson, 1969]. This mobile robot performed some simple tasks in the impoverished world of a university laboratory [McCorduck, 1979].

[†] This table was constructed by referring to several sources [McCorduck, 1979; Barr et al., 1982; Miller, 1983; Saveriano, 1988; Groover, 1988; Brooks, 1991b; Brahan, 1992; Freedman, 1993]. All publications noted in the table have been italicized. The titles of periodic publications are located beside the first year of printing.

[‡] An outline of work in neural networks and robotics is given by [Horne et al., 1990], while a current view of the history of and research in neural networks is located in [Jubak, 1992].

A general observation regarding the relationship between the field of AI and both streams of robotics is that most ideas developed in the field of AI are encountered with some variable time delay in the field of robotics. This observation is not exclusive to the field of robotics, since researchers from several fields view the tools and techniques of AI as the appropriate means to develop intelligence. The research on machine vision in AI began in the mid 1960s, while the implementation of industrial machine vision first occurred in the late 1970s. The research on expert systems in the 1960s and 1970s led to commercial expert systems in the 1980s. The conference on *Robotics and Expert Systems* was held in the following four years, 1985, 86, 87, 89. Learning about the limitations of expert systems, such as the bottleneck in knowledge acquisition and the brittleness of expert systems outside of their domain of expertise, shifted the research to issues concerning knowledge and ways to overcome this brittleness.

Two key developments in AI, which are somewhat related to these realizations, have been taking place in the 1980s and 1990s. The first key development was the launching of several large-scale projects on AI and intelligent computing [Bensen, 1986]. The Japanese led by establishing the fifth generation project in 1982. This national strategy was responded to by Europe and the United States. In 1984, several major corporations in the United States initiated the Cyc project. The purpose of this ten year project was to build a large knowledge base with common sense by building up facts on what has been termed consensus reality; thereby, overcoming the brittleness of expert systems [Lenat and Guha, 1990]. A very preliminary discussion of applying such large knowledge bases to robotics was remarked upon in [Basran, 1991]. Second, the research on qualitative reasoning, naive physics, and qualitative physics has become established well enough to produce a comprehensive summary in [Weld and De Kleer, 1990]. To what degree these latest developments will affect intelligent robotics is yet to be discovered.

On the other hand, the tough problems of working with the real, physical world of robotics can serve as a reminder to researchers in AI that working with abstracted systems in a computer may be far from useful. In fact, the history of robotics focusses, for the most part, on physical problems, while the history of AI tends to focus on abstracted problems.

Current Paradigms

In this section, we briefly describe the viewpoints of certain current paradigms of intelligent robotic systems. Three previously mentioned ideas should be kept in mind while reading this section. These ideas are the architecture of intelligence, the simple framework for an intelligent robotic system, and the means used to test an intelligent robotic system.

Control Community

Recently, the paradigm of intelligent control was defined as the discipline which lies at the intersection of AI, operations research, and control [Saridis, 1983]. The basic motivation of this community has been to develop an analytic approach to the study of intelligent machines [Saridis, 1983; Meystel, 1988a, 1988b; Tzafestas, 1991; Valavanis and Saridis, 1992]. The general theory of intelligent machines is founded on the principle of increasing intelligence with decreasing precision and was formulated with a careful definition of several terms such as machine knowledge and machine intelligence [Valavanis and Saridis, 1992]. The concept of entropy from information theory unifies this theory as the measure of performance. Based upon this general theory for intelligent machines, a mathematical model of an intelligent robotic system, a specific type of intelligent machine, was constructed [Valavanis and Saridis, 1992]. This mathematical model of an intelligent robotic system is based upon a hierarchical architecture with three levels. Starting from the top of the hierarchy, these levels are organization, coordination, and execution. Perception is connected to action at the top level of this hierarchy. This model of an intelligent robotic system was investigated by simulating the operation of a mobile robot in a nuclear plant and an industrial robot involved in assembly [Valavanis and Saridis, 1992]. This approach is also being explored with actual robotic systems intended for space operation [Desrochers, 1992].

AI Community

The previous inquiry into the relationship between AI and robotics discussed the ambiguity between these fields. It also contended that if there is a scale from physical problems to abstracted problems, then robotics from an AI perspective has a proclivity for abstracted problems. A recent robotics competition for indoor mobile robots that was sponsored by the American Association for AI supports this point [Dean and Bonasso, 1993]. The tasks for the competition were chosen for their particular interest to the AI community. These tasks were navigating in a cluttered environment without colliding into static obstacles and moving people, exploring the environment to find ten objects, and executing a command to visit three objects in a specific order and then return home. The environment consisted of foam-core sheets for the walls, cardboard boxes for the obstacles, and eight foot, vertical, polyvinyl chloride, drainage pipes for the objects. These pipes or poles could be rigged with whatever researchers chose to aid the robot in finding the objects while exploring the environment. The tasks and design of environment seem strikingly similar to [Nilsson, 1969]. In the same magazine that discussed this competition, Gat in [Gat, 1993] responded in part to the challenge presented to the AI community by [Brooks, 1991a, 1991b]. Of the two declared winners for this indoor mobile robotics competition [Congdon et al., 1993], one winner had a hierarchical architecture while the other had a distributed architecture. For a thorough discussion of different architectures for intelligent mobile agents and a proposal for a new one, refer to [Ferguson, 1992].

Robotics Community

The problems of coping with the physical world are stressed by the field of robotics. This commonality has been expressed by different researchers. For example, robotic systems are physical processors [Lee, 1989], the physical grounding hypothesis is necessary to build intelligent robotic systems [Brooks, 1990], and robotic systems are physical problem-solving systems [Thorpe, 1990]. In the work of [Brooks, 1986a, 1991a, 1991b], the architecture of intelligence is the subsumption architecture and the intelligent connection of perception to action is a direct one with no traditional representation or reasoning. The physical test for intelligence is that a mobile robot should be able to carry out its tasks in an environment which may be engineered for humans, for example indoor environments, but without further restrictions or modifications to help the mobile robot. A review of the work on mobile robotics from this perspective is given in [Brooks and Flynn, 1989]. Meanwhile, Thorpe recognized the dependence between the performance of a mobile robotic system and its perceptual systems in [Thorpe et al., 1991]. According to this reference, perception is the key bottleneck of progress. In [Lee, 1989], this idea was aptly expressed by stating that the limits of intelligence are determined by the quality and effectiveness of the perceptual systems. On the other hand, if the goal of the researcher(s) is not to build an autonomous robotic system, but a robotic system where interaction with humans is more important, then a greater effort and emphasis is placed on the intelligence of the human interface [Schenker, 1991; Schweitzer, 1991].

The recent idea of computer integrated manufacturing has affected the meaning of applying AI techniques in a manufacturing environment. The intelligence of a particular industrial robot or AGV is not as important as the intelligence of the operation of a robotic based workcell, or of the operation of a group of workcells, or of the overall operation of certain aspects of a computer integrated factory [Cook and Emrich, 1987; O'Grady et al., 1989; O'Hare, 1989; Swyt, 1989; Shapiro, 1992]. In some cases, the work in AI and manufacturing has become abstract enough to question its practical utility.

Modelling and Simulation Community

The latest entrant with publications on intelligent agents is from the modelling and simulation community and is represented by the following work [Zeigler, 1990a, 1990b; Zeigler et al., 1989]. The work of Zeigler in the field of modelling and simulation was influenced by research in system theory and is based primarily upon the discrete event formalism. This author uses the term, high autonomy systems, rather than intelligent systems because he maintains that autonomy is a better descriptor than intelligence since the goal of achieving autonomy implies having the needed intelligence [Zeigler, 1990b]. He proposes an architecture for high autonomy systems, which is a combination of the three level hierarchical control architecture of [Saridis, 1983] and his own approach to modelling developed for discrete event systems. The result is a model based architecture, where models are viewed as a form of knowledge. The three layers of this architecture are called management, coordination, and execution. The issue of model base management for intelligent agents is a particular emphasis of this research. The focus of the simulations of intelligent robotic systems in this work have been on the task-related cognitive requirements, which clearly makes this research fall within the purview of the AI community.

Conclusion

The intent of this paper was to examine how ideas on intelligence and the field of robotics have intersected to synthesize intelligent robotics. The framework of paradigms was used in this analysis. The paradigm of a researcher plays a pivotal role in determining which ideas and which realities meet on the playing field of intelligent robotics. Certain broad perspectives on intelligent robotics systems were presented to support this claim.

Progress in computing and architectures for computing continue to change the type of problems that can be potentially probed. The possibilities of applying parallel architectures, including neural networks, to control problems has been investigated [Fizany and Bejczy, 1992]. Architectures which are hybrids of symbolic, neural, and fuzzy computing are discussed in [Kandel and Langholz, 1992]. The question of what is the best architecture to develop an intelligent robotic system continues to be answered in many different ways, for examples see [Chen and Trivedi, 1992; Connell, 1992; Liscano et al., 1992; Shibata et al., 1992]. Maybe the question should be what is the best architecture for a particular problem.

Acknowledgements

The first author thanks the Autonomous Systems Laboratory of the Institute for Information Technology at the National Research Council of Canada for providing the excellent resources to write this paper. He also thanks the Natural Sciences and Engineering Research Council of Canada and the Ontario Graduate Scholarship program for the graduate student funding that they have provided me. He also expresses his appreciation to Elizabeth Stuck and Fazel Famili for providing important references and for reviewing this paper.

References

Albus, James S., [1991], "Outline for a Theory of Intelligence", IEEE Trans. on Syst., Man, and Cybern., Vol. 21, No. 3, pp. 473-509.

Albus, J., McCain, H., and Lumia, R., [1987], *NBS Standard Reference Model for Telerobot Control System Architecture (NASREM)*, National Bureau of Standards Technical Note 1235.

Armer, Paul, [1960], "Attitudes toward Intelligent Machines", *Computers and Thought*, editors E.A. Feigenbaum and J. Feldman, McGraw-Hill, 1963, pp. 389-405.

Barr, Avron, Cohen, Paul R., and Feigenbaum, Edward A., editors, *The Handbook of Artificial Intelligence*, Heuristech Press, 1982, Volumes 1, 2, 3.

Basran, Jagdeep S., [1991], "Ontology and Robotics", report submitted for course in Knowledge Engineering taken at the University of Ottawa.

Bensen, Ian, editor, [1986], *Intelligent Machinery: Theory and Practice*, Cambridge University Press.

Birk, John R. and Kelley, Robert B., [1981], "An Overview of the Basic Research Needed to Advance the State of Knowledge in Robotics", *IEEE Trans. on Syst., Man, and Cybern.*, Vol. 11, No. 8, pp. 574-9.

Brady, Michael, [1985], "Artificial Intelligence and Robotics", *Artificial Intelligence*, Vol. 26, pp. 79-121.

Brahan, J.W., [1992], "Artificial Intelligence and the Real World", *Proc. of the 6th Int. Conf. on Syst. Research, Informatics, and Cybern.*, Vol. 2, pp. 1-6.

Brooks, Rodney A., [1986a], "A Robust Layered Control System for a Mobile Robot", *IEEE J. of Robot. and Autom.*, Vol. 2, No. 1, March 1986, pp. 14-23.

Brooks, Rodney A., [1986b], "Achieving Artificial Intelligence through Building Robots", Mass. Inst. of Technol., A.I. Memo 899, May 1986.

Brooks, Rodney A. and Flynn, Anita M., [1989], "Robot Beings", *Proc. of the 1989 IEEE/RSJ Int. Conf. on Intell. Robots and Syst.*, pp. 2-10.

Brooks, Rodney A., [1990], "Elephants Don't Play Chess", *Robot. and Auton. Syst.*, Vol. 6, pp. 3-15.

Brooks, Rodney A., [1991a], "Intelligence without Representation", *Artificial Intelligence*, Vol. 47, Nos. 1-3, pp. 139-60.

Brooks, Rodney A., [1991b], "Intelligence without Reason", Mass. Inst. of Technol., A.I. Memo 1293, April 1991.

Chen, ChuXin and Trivedi, Mohan M., [1992], "Architecture and Automatic Task Planning for Integrated Sensor-Based Robots", *Proc. of the 1992 IEEE/RSJ Int. Conf. on Intell. Robots and Syst.*, pp. 925-32.

Congdon, Clare, et al., [1993], "CARMEL versus FLAKEY A Comparison of Two Winners", *AI Magazine*, Vol. 14, No. 1, Spring 1993, pp. 49-57.

Connell, Jonathan H., [1992], "SSS: A Hybrid Architecture Applied to Robot Navigation", *Proc. of the 1992 IEEE/RSJ Int. Conf. on Intell. Robots and Syst.*, pp. 2719-24.

Cook, James H. and Emrich, Mary, [1987], "Artificial Intelligence in Manufacturing", *Handb. of Manuf. High Technol.*, pp. 741-64.

Dean, Thomas and Bonasso, R. Peter, [1993], "AAAI Robot Exhibition and Competition", *AI Magazine*, Vol. 14, No. 1, Spring 1993, pp. 34-48.

Desrochers, Alan E., editor, [1992], *Intelligent Robotic Systems for Space Exploration*, Kluwer Academic Publishers.

Ferguson, Innes A., [1992], *TouringMachines: An Architecture for Dynamic, Rational, Mobile Agents*, Doctoral Dissertation, University of Cambridge.

Fizany, A. and Bejczy, A., editors, [1992], *Parallel Computation Systems for Robotics: Algorithms and Architectures*, World Sci. Publishing Co.

Freedman, Paul, [1993], "Robotics in the 1990's", *IEEE Canadian Review*, Winter / hiver 1993, pp. 5-8.

Gat, Erann, [1993], "On the Role of Stored Internal State in the Control of Autonomous Mobile Robots", *AI Magazine*, Vol. 14, No. 1, Spring 1993, pp. 64-73.

Gevarter, William B., [1982], *An Overview of Artificial Intelligence and Robotics Volume II - Robotics*, prepared for NASA, NBSIR 82-2479.

Gevarter, William B., [1983], *An Overview of Artificial Intelligence and Robotics Volume I - Artificial Intelligence, Part A, Part B, Part C*, NASA TM - 85836, 85838, 85839.

Gevarter, William B., [1985], *Intelligent Machines: An Introductory Perspective of Artificial Intelligence and Robotics*, Prentice-Hall.

Groover, Mikell P., [1988], "Automation", *Int. Encycl. of Robot.: Applic. and Autom.*, Richard C. Dorf, editor in chief, Vol. 1, pp. 136-51.

Horne, Bill, Jamshidi, M., and Vadiee, Nader, [1990], "Neural Networks in Robotics: A Survey", *J. of Intell. and Robot. Syst.*, Vol. 3, No. 1, pp. 51-66.

Hunt, V. Daniel, [1985], *Smart Robots - A Handbook of Intelligent Robotic Systems*, Chapman and Hall.

Inoue, Hirochika, [1985], "Building a Bridge Between AI and Robotics", *Proc. of 9th Int. Joint Conf. on AI*, August 1985, p. 1231-37.

Jubak, Jim, [1992], *In the Image of the Brain: Breaking the Barrier between the Human Mind and Intelligent Machines*, Little, Brown and Company.

Kandel, Abraham and Langholz, Gideon, editors, [1992], *Hybrid Architectures for Intelligent Systems*, CRC Press.

Kirsh, D., [1991a], "Foundations of AI: the big issues", *Artificial Intelligence*, Vol. 47, Nos. 1-3, pp. 3-30.

Kirsh, D., [1991b], "Today the earwig, tomorrow man?", *Artificial Intelligence*, Vol. 47, Nos. 1-3, pp. 161-84.

Klir, George J., [1985], *Architecture of Systems Problem Solving*, Plenum Press.

Kuhn, T.S., [1962], *The Structure of Scientific Revolutions*, University of Chicago Press.

Lachman, Roy, Lachman, Janet L., and Butterfield, Earl C., [1979], *Cognitive Psychology and Information Processing: An Introduction*, Lawrence Erlbaum Associates.

Lee, Mark H., [1989], *Intelligent Robotics*, Halstead Press.

Lenat, Douglas B. and Guha, R.V., [1990], *Building Large Knowledge-Based Systems: Representation and Inference in the Cyc Project*, Addison-Wesley Publishing Co.

Liscano, R., Fayek, R.E., and Karam, G.M., [1992], "A Blackboard, Activity-Based Control Architecture for a Mobile Platform", *Proc. of the 1992 IEEE/RSJ Int. Conf. on Intell. Robots and Syst.*, pp. 333-8.

McCorduck, Pamela, [1979], *Machines who Think*, W.H. Freeman and Company.

Meystel, A., [1988a], "Intelligent Control: A Sketch of the Theory", *J. of Intell. and Robot. Syst.*, Vol. 1, No. 1, pp. 97-107.

Meystel, A., [1988b], "Intelligent Control in Robotics", *J. of Robot. Syst.*, Vol. 5, No. 4, pp. 269-308.

Miller, Richard K., [1983], *Intelligent Robots*, SEAI Institute and Technical Insights Inc.

Mowforth, P. and Bratko I., [1986], *AI and Robotics: Flexibility and Integration*, The Turing Institute, TIRM-86-014.

Newell, A. and Simon, H.A., [1972], *Human Problem Solving*, Prentice-Hall.

Nilsson, N. G., [1969], "A mobile automaton: An application of artificial intelligence techniques", *Proc. of 1st Int. Joint Conf. on AI*, pp. 509-20.

Nitzan, David, [1985], "Development of Intelligent Robots: Achievements and Issues", *IEEE J. of Robot. and Autom.*, Vol. 1, No. 1, pp. 3-13.

Norman, Donald A., [1991], "Approaches to the study of intelligence", *Artificial Intelligence*, Vol. 47, Nos. 1-3, pp. 327-46.

O'Grady, P.J., Bao, H., and Lee, K.H., [1987], "Issues in Intelligent Cell Control for Flexible Manufacturing Systems", *Comp. in Ind.*, Vol. 9, No. 1, pp. 25-36.

O'Hare, G.M.P., [1989], "Designing Intelligent Manufacturing Systems: A Distributed Artificial Intelligence Approach", *Intell. Manuf. Syst. 1989*, pp. 17-25.

Rich, Elaine and Knight, Kevin, [1991], *Artificial Intelligence*, McGraw-Hill Inc.

Rosenbloom, Paul S., et al., [1991], "A preliminary analysis of the Soar architecture as a basis for general intelligence", *Artificial Intelligence*, Vol. 47, Nos. 1-3, pp. 289-325.

Saridis, George N., [1983], "Intelligent Robotic Control", *IEEE Trans. on Autom. Control*, Vol. 28, No. 5, pp. 547-57.

Saveriano, Jerry W., [1988], "Pioneers of Robotics", *Int. Encycl. of Robot.: Applic. and Autom.*, Richard C. Dorf, editor in chief, Vol. 2, pp. 1078-92.

Schenker, Paul S., [1991], "Intelligent Robotics for Space Applications", *Intelligent Robotic Systems*, Spyros Tzafestas, editor, Marcel Dekker, pp. 545-91.

Schultz, Emily A. and Lavenda, Robert H., [1990], *Cultural Anthropology: A Perspective on the Human Condition*, West Publishing Company.

Schweitzer, G., [1991], "The Robot as an Intelligent Interactive Machine", *Mechatronics: Mechanics, Electronics, Control*, Vol. 1, No. 4, pp. 525-33.

Shapiro, Stuart C., Editor in Chief, [1992], "Manufacturing, AI in", *Encycl. of AI* , John Wiley & Sons, Second Edition, Vol. 2, pp. 903-9.

Shibata, Takanori, et al., [1992], "Hybrid Symbolic and Neuromorphic Control for Hierarchical Intelligent Control", *Proc. of the 1992 IEEE/RSJ Int. Conf. on Intell. Robots and Syst.*, pp. 2081-6.

Staugaard, Andrew C., [1987], *Robotics and AI: An Introduction to Applied Machine Intelligence*, Prentice-Hall.

Swyt, Dennis A., [1989], "AI in Manufacturing: The NBS*AMRF as an Intelligent Machine", *Robot. and Auton. Syst.*, Vol. 4, pp. 327-332.

Tapscott, Don and Caston, Art, [1993], *Paradigm Shift: The New Promise of Information Technology*, McGraw-Hill Inc.

Thorpe, Charles E., editor, [1990], *Vision and Navigation The Carnegie Mellon Navlab*, Kluwer Academic Publishers.

Thorpe, Charles, et al., [1991], "Toward Autonomous Driving: The CMU Navlab - Part I Perception", *IEEE Expert*, August 1991, pp. 31-42.

Turing, A., [1948], "Intelligent Machinery", *Machine Intelligence 5*, editors Meltzer, Bernard and Michie, Donald, American Elsevier Publishing Co. Inc., 1970, pp. 3-23.

Turing, A., [1950], "Computing Machinery and Intelligence", *Computers and Thought*, editors E.A. Feigenbaum and J. Feldman, McGraw-Hill, 1963, pp. 11-35.

Tzafestas, Spyros, [1991], "Introduction to Intelligent Robotic Systems", *Intelligent Robotic Systems*, Spyros Tzafestas, editor, Marcel Dekker, pp. 3-42.

Valavanis, Kimon P. and Saridis, George N., [1992], *Intelligent Robotic Systems: Theory, Design, and Applications*, Kluwer Academic Publishers.

Weld, Daniels S. and De Kleer, Johan, editors, [1990], *Qualitative Reasoning about Physical Systems*, Morgan Kaufmann Publishers.

Woods, P.E., [1989], "Defence Robotics in Canada", *Proc. of the 2nd Workshop on Military Robot. Applic.*, August 1989, pp. 1-3.

Zeigler, Bernard, [1990a], *Object-Oriented Simulation with Hierarchical, Modular Models: Intelligent Agents and Endomorphic Systems*, Academic Press, New York, USA.

Zeigler, Bernard P., [1990b], "High Autonomy Systems: Concepts and Models", *Proc. of AI, Simul., and Planning in High Autonomy Syst.*, pp. 2-7.

Zeigler, Bernard P., Cellier, Francois E., and Rozenblit, Jerzy W., [1989], "Design of a Simulation Environment for Laboratory Management by Robot Organizations", *J. of Intell. and Robot. Syst.*, Vol. 1, No. 1, pp. 299-309.

902219

Système à base de connaissances pour la planification des tâches avec applications en robotique

Michel Corneliu Murgulescu
Ecole Polytechnique de Montréal, Département de génie électrique et génie informatique, C.P. 6079, succ. A
Montréal (Québec), Canada H3C 3A7

#149675

Résumé

La planification des tâches par générateur de plans d'actions, thème de recherche classique en intelligence artificielle, se rapporte en robotique à la commande des robots. L'objet des plans est de donner aux robots les moyens de décider eux-mêmes des actions physiques à exécuter, pour atteindre des objectifs précis. Ces moyens consistent essentiellement en un programme de résolution des problèmes, appelé "générateur de plans d'actions", complété parfois par un autre programme pour contrôler l'exécution des plans engendrés.

L'approche utilisée est différente de celle que l'on rencontre habituellement. La différence est qu'ici nous considérons le générateur de plans comme un système expert capable de planifier les tâches d'un système robotisé à partir des informations envoyées par un système de capteurs. Actuellement, le système travaille sur la base d'un dialogue avec l'usager. Dans une étape ultérieure d'implantation opérationnelle, le "dialogue" s'établira entre le système expert et le module de perception du système robotisé.

L'objectif principal de ce projet est de généraliser les résultats pour réaliser une coquille de système expert. Cet outil paramétrable permettra la planification des tâches d'un système robotisé en fonction de son domaine d'application.

1. Introduction

1.1 Enoncé du problème

Le système expert que nous avons réalisé a pour but la planification des tâches d'un système robotisé. Le système robotisé considéré, est un système de recherche et distribution des appareils dans un laboratoire d'asservissements, désigner par l'abréviation SRRD (Système Robotisé de Recherche et Distribution). Il a pour tâche de chercher les appareils appartenant à une manipulation que l'on désire exécuter et de les distribuer sur des tables de travail (il s'agit des manipulations de laboratoire dans un milieu universitaire).

Hypothèses:

Les appareils existent en plusieurs exemplaires et sont rangés dans deux armoires sur différentes tablettes, chacune pouvant contenir plusieurs rangées d'appareils. Une table auxiliaire permet de ranger les appareils en attente. Un système de vision équipant le robot permet de "voir" une tablette et d'identifier les appareils situés sur la premiére rangée.

Nous avons tout d'abord décomposé le problème en deux phases: la première concerne la recherche des appareils (appartenant à une certaine manipulation) dans deux armoires; la deuxième concerne la distribution des appareils sur les tables de travail.

1.2 Formalisation du problème

Nous allons décrire la procédure d'identification et les phases nécessaires pour accomplir la tâche du SRRD. Les appareils sont identifiés par deux codes, un pour indiquer l'appartenance à une manipulation et l'autre pour indiquer l'appartenance à une catégorie d'appareils (moteurs, voltmètres, amplificateurs, etc.). En plus des codes, nous avons associé à chaque catégorie d'appareil, un attribut qui permettra au système de vision d'identifier leur gabarit. Ainsi il y a trois types d'appareils: grand, moyen et petit.

Les moyens d'identification s'établissent comme suit:

- Le premier code (géométrique) permet d'établir si l'appareil appartient à une certaine manipulation pour satisfaire une requête du système. Il sera nécessaire dans la phase de recherche des appareils;
- Le deuxième code (numérique) permet de trouver l'appartenance d'un appareil à une categorie. Il sera nécessaire dans la phase de distribution des appareils pour vérifier si tous les appareils ont été distribués et pour éviter d'avoir deux appareils de la même catégorie sur la même table;
- Le troisième moyen d'identification (attribut) permet l'application de règles de production. Il sera nécessaire dans la phase de recherche des appareils pour pouvoir charger le plateau du robot.

Le système de reconnaissance visuelle sera remplacé par un système de reconnaissance de formes à travers un réseau neuronal dans une version plus évoluée. Les signaux d'entrée seront directement traités par le système et permettront une plus grande homogénéité dans les échanges d'informations entre le système de vision et le système expert (interprétation de l'environnement, décisions et exécution des plans). La rapidité d'exécution y gagnera beaucoup.

Décomposition de la tâche du SRRD en deux phases:

- La phase de recherche des appareils qui aura pour but de chercher dans les deux armoires tous les appareils appartenant à une certaine manipulation. Cette phase peut être décomposée en deux étapes:
  1) Identification des appareils appartenant à la manipulation qu'on veut exécuter;
  2) Chargement du plateau du robot en vue de transporter et de distribuer les appareils.
- La phase de distribution des appareils qui aura pour but de distribuer sur chaque table de travail tous les appareils nécessaires à l'exécution de la manipulation. Cette phase implique deux étapes d'identification: une première identification pour la phase de recherche et une deuxième pour répondre à une question du type: "cette table est-elle complete?"

Les phases et les étapes nécessaires à l'accomplissement de la tâche du système robotisé sont représentées sur la Fig.1.

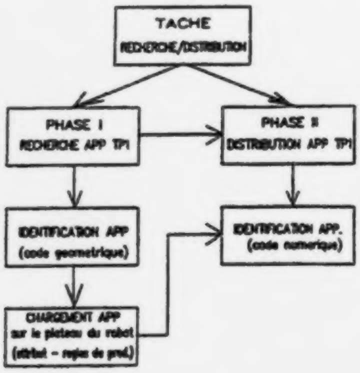


Fig.1 Les phases et les étapes pour acomplir la tâche du SRRD

2. Formalisation mathématique du problème

Pour une résolution homogène de notre problème nous allons le formaliser mathématiquement ce qui permettra une vision globale et structurée. Le système opère sur l'espace des états constitué par les états initiaux pour arriver à des états buts (finaux). Pour réaliser cette opération il utilise une série d'actions appartenant à l'espace des actions (Fig.2).

Les entrées du système seront les états initiaux et les sorties (les états buts désirés) seront des listes représentant des enchaînements des actions capables de faire passer le système d'un état à un autre. L'espace des états initiaux est constitué par l'ensemble des états des appareils sur les tablettes des armoires et par l'ensemble des états intermédiaires des appareils sur les tables de travail. Finalement, l'espace des actions est constitué par l'ensemble des actions nécessaires pour faire passer le système d'un état à un autre. Voici les actions exécutées par le système, nécessaires pour passer d'un état à un autre.

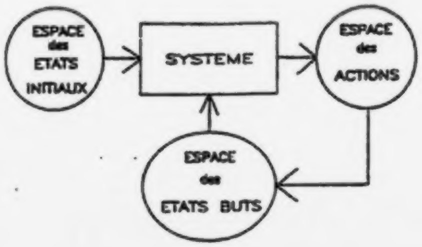


Fig.2 Les entrées et les sorties du système robotisé

IDENTIFICATION_APP action exécutée par le système de vision du SRRD

PREND_APP
DEPOSE_APP_PLATEAU actions exécutées par le
DEPOSE_APP_TAB_AUX bras manipulateur du SRRD
DEPOSE_APP_TAB

DEPLACE_ROBOT_TABLE actions exécutées par le
RETOURNE_ROBOT_ARM système de navigation

Du point de vue mathématique une action est une fonction qui permet de passer d'un état à un autre; un état est obtenu par l'application d'une fonction (action) à un état précédent.

$$\begin{aligned} \text{Action}(i) &= F_1\,[e(i),\ \ e(i+1)\,] \\ e(i+1) &= F_2\,[e(i),\ \text{Action}(i)] \end{aligned}$$

Par exemple l'action "DEPOSE_APP_PLATEAU" est appliquée à l'état "APP_pince" pour obtenir l'état "APP_PLATEAU". Aussi, l'état "APP_TABLE" est obtenu de l'état "APP_PLATEAU" par l'application de l'action "PREND_APP" suivie de l'action "DEPOSE_APP_TABLE". Il faut remarquer que l'action "PREND_APP" est toujours suivie d'une action "DEPOSE_APP_..." étant donné que le bras manipulateur doit être libéré pour une autre action. Son état doit être "LIBRE-pince". La formalisation mathèmatique permet l'encadrement du problème avec un certain degré de généralité.

3. Le scénario d'exécution de la tâche

Le Système Expert pour la Planification des Tâches, abrégé par SEPT doit reproduire un scénario de recherche et de distribution des appareils, en fonction de certaines hypothèses:

a) La camera peut identifier seulement les appareils qui se trouve à "vue", c'est-a-dire, sur la première rangée de l'armoire. Pour identifier les appareils "cachés" sur une rangée arrière, le robot doit enlever d'abord les appareils qui cachent les premiers;

b) Pour la distribution, le système de vision doit identifier les appareils existant sur la table de travail, mémoriser et comparer la scène avec une "scène standard" (table complète). Alors, le système sera capable d'ajouter seulement les appareils manquants;

c) Les appareils sont distribués en "vrac" sur les tables de travail.

SCENARIO:

Le SRRD reçoit une commande: "Chercher et distribuer les appareils pour la manipulation No." Suite à cette commande, le système va enchaîner une série d'actions:

Action 1 - Le SRRD "regarde" l'armoire avec son système de vision jusqu'au moment ou il identifie un appareil appartenant à la liste des appareils pour la manipulation No.

Action 2 - Si parmi les appareils de la première rangée il n'y a aucun qui soit dans la liste, il "prend" un appareil, "dépose" l'appareil sur la table auxiliaire pour pouvoir "regarder" la deuxième rangée (et recommence autant de fois que nécessaire)

Action 3 - Une fois identifié l'appareil appartenant à la liste, le robot "prend" l'appareil et le "dépose" sur son plateau

Action 4 - Le système exécute le cycle: regarde-prend-dépose jusqu'à ce que le plateau du robot soit plein
Action 5 - Le SRRD se "deplace" jusqu'à une table de travail
Action 6 - Le SRRD "regarde" la table de travail pour établir s'il y a des appareils sur la table. S'il n'y en a pas il passe à l'action suivante. S'il y en a, soit il doit passer à la table suivante si la table est complète, soit il "dépose" sur la table les appareils qui manquent
Action 7 - Le SRRD "dépose" les appareils sur la table de travail
Action 8 - Le SRRD "retourne" à l'armoire avec le plateau vide et recommence le cycle
Action 9 - Le SRRD "répète" le cycle d'actions, de l'action 1 à l'action 8 autant de fois que nécessaire pour que les tables de travail soient complètes
Action 10 - Le SRRD "arrête"jusqu'à une nouvelle commande.

4. La représentation de la connaissance

Les trois éléments principaux d'un système expert sont:

- La base de connaissance constituée par la base de faits et la base de règles. Elle contient les jugements, l'intuition, l'expérience et la connaissance de l'expert;
- Le mécanisme d'inférence pour réaliser l'interprétation de la connaissance, les inférences logiques et les modifications dans la base de connaissance. Il représente le raisonnement du système expert;
- Le mécanisme de contrôle constitué par les stratégies de contrôle du processus d'inférence.

En général, trois formalismes sont utilisés pour la représentation de la connaissance: les règles de production, les objets structurés et la logique des prédicats. Ces formalismes sont souvent désignés par le terme "Pattern Directed Inference Systems" [12] soit, systèmes d'inférence dirigés par les filtres.

Nous avons adopté comme formalisme de représentation de la connaissance les règles de production. C'est le plus utilisé et le mieux adapté à notre problème.

Il a été utilisé dans la théorie des automates, des grammaires formelles et des langages de programmation avant d'être utilisé dans les modèles psychologiques et les systèmes expert.

Nous avons choisi ce mode de représentation parce qu'il permet aussi de créer des fragments de connaissance indépendants les uns des autres, donc facilement modifiable. Ses avantages: la lisibilité, la souplesse, la modularité des règles et la possibilité d'introduire des coefficients de vraissemblance permettant de pondérer les connaissances.

5. Le mécanisme d'inférence et son contrôle

La base de connaissance, peu importe le formalisme utilisé pour la bâtir, constitue la perception que le système possède concernant son environnement de travail. Le moteur d'inférence permettant le fonctionnement du système expert, est alimenté par la base de connaissance. Il construit dynamiquement le raisonnement et décide quelles sont les règles à déclencher et dans quel ordre.

Etant donné que nous avons utilisé une "coquille" (VP-EXPERT) pour développer le prototype du système expert, le mécanisme de contrôle vienne avec l'outil de développement.

Quel que soit le mode de raisonnement utilisé, le moteur d'inférence possède un cycle de base qu'on peut définir comme des cycles de travail enchaînés pour aboutir au résultats; le cycle de base comprend quatre phases:

- Phase de sélection réalisée sur un sous-ensemble de la base connaissance qui méritent plus d'attention que le reste de la base. Ce choix tient d'une stratégie du moteur qui décide qu'un tel groupe de règles est privilégié par rapport à un autre; soit en fonction du problème, soit en fonction de l'introduction de méta-règles;
- Phase de filtrage c'est l'étape clé du cycle de base du moteur d'inférence qui consiste à comparer la partie prémisse (en chaînage avant) ou la partie conclusion des règles (en chaînage arrière), avec les faits de la base (en chaînage avant) ou avec l'ensemble des sous-buts (en chaînage arrière) pour déterminer l'ensemble

des règles applicables;

- Phase de résolution de conflits pour choisir la règle qui va être appliquée effectivement. Pour cette phase VP-EXPERT utilise le chaînage arrière;
- Phase d'exécution qui consiste dans l'application de la règle choisie précédemment. L'application de la règle peut avoir plusieurs effects: modification de la base de faits, appel de procédures externes, etc. Dans la phase d'exécution VP-EXPERT utilise le backtracking, le filtrage et la génération de faits suivie de tests. L'arrêt du cycle de base du moteur d'inférence dépend du mode de raisonnement utilisé.

6. L'outil de développement VP-EXPERT

Pour la phase de prototypage du système, le choix d'une coquille de système expert s'imposait pour au moins deux raisons:

1. D'abord elle permet une modélisation rapide du système proposé et une vérification quasi immédiate de l'homogénéité de la connaissance insérée dans le système. Le prototype permettra donc, à moindre coût (argent et temps), de déterminer la faisabilité du projet en fonction des options choisies.
2. Ensuite, une coquille de système expert est généralement dotée de systèmes de contrôle élaborés qui permettent un passage au crible minutieux du prototype et un recensement, que l'on souhaite le plus exhaustif possible. Il ne faut cependant pas perdre de vue que le prototype n'est qu'une phase transitoire. Il convient donc de maîtriser les mécanismes d'inférence et de contrôle afin de pouvoir les reproduire sur le système réel. Les critères qui nous ont permis de choisir la coquille VP-EXPERT sont tirés du [13].

6.1 Les caractéristiques de VP-EXPERT

Dans le cas particulièr du logiciel VP-EXPERT, la représentation de la connaissance, le contrôle, l'environnement et les utilitaires se présentent de la manière suivante:

6.1.1 Représentation des connaissances

Comme dans tout système expert, les connaissances sont représentées sous forme de faits et de règles. En ce qui concerne les règles, autant pour la conclusion directe que pour l'alternative, on peut utiliser un coefficient de vraissemblance. Avec PROLOG, les coefficients de vraissemblance deviendront de véritables probabilités qu'il sera possible de manipuler afin d'obtenir une modélisation pragmatique et tangible du monde réel.

Les règles, comme pour PROLOG, fonctionnent selon la logique formelle: prémisse -> conclusion. La prémisse peut être une conjonction ou une disjonction de clauses. La conclusion de la règle peut être une liste d'une ou plusieurs expressions de la forme: VARIABLE = VALEUR avec coefficient de vraissemblance et conclusion alternative. Il est possible d'ajouter un texte explicatif avec l'option PARCE QUE.

6.1.2 Le contrôle

VP-EXPERT travaille en chaînage arrière ou en chaînage avant. Le moteur d'inférence cherche à évaluer un but qu'il identifie dans le bloc "ACTIONS" du programme (le bloc "ACTIONS" joue le rôle du programme principal) comme étant l'argument d'une clause FIND. Une fois le but identifié, il filtre toutes les règles qui ont ce but dans leur conclusion et il développe les sous-buts en profonduer d'abord (depth-search). Il applique, sauf s'il arrive à une certitude, toutes les règles qui peuvent s'appliquer pour renforcement de l'évidence (breadth-search). Donc la recherche est exhau stive. S'il ne peut pas établir la valeur d'un but, il cherchera dans la partie "STATEMENTS" du programme une instruction ASK qui lui permettra d'acquérir la valeur en question.

6.1.3 L'environnement et les utilitaires

Le choix d'une coquille de système expert pour un développement de prototype prend toute sa mesure lorsque l'on considère les outils à notre disposition pour le contrôle de l'exécution:

- Explication des raisonnements: POURQUOI? (WHY?)

et COMMENT? (HOW?) pendant la consultation;
- Traceur pour l'ensemble du raisonnement avec option SLOW pour suivre aisément l'exécution du programme
- Explication des questions, commentaires et AIDE (HELP) à plusieurs niveaux;
- Modification des réponses en cours de traitement option WhatIf);
- Interface utilisateur multifenêtre permettant de suivre: le dialogue avec le système, la trace par l'affichage des règles et la trace par affichage des résultats;
- Une autre possibilité très intéressante est la génération d'un fichier contenant le cheminement de l'information (path information) pendant la session de consultation. On peut visualiser le contenu de ce fichier soit sous forme d'arbre graphique (graphics tree) soit sous forme de texte (text tree);
- Une autre possibilité importante du VP-EXPERT est l'interfaçage avec d'autres logiciels. Elle permet de contourner une des faiblesses des logiciels d'Intelligence Artificielle, la manipulation des données numériques.

6.1.4 La structure d'un programme VP-EXPERT

Un programme écrit à l'aide du VP-EXPERT doit avoir une structure précise: un bloc ACTIONS suivi d'un bloc RULES et à la fin du programme les instructions (STATEMENTS), voir Fig.3.

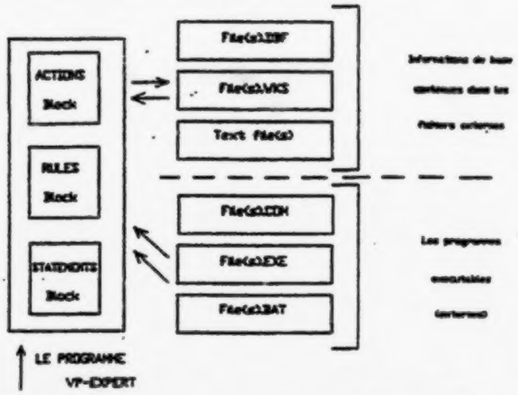


Fig.3 La structure d'un programme VP-EXPERT

Le bloc ACTIONS joue le rôle du programme principal. Il contient le keyword "ACTIONS" suivi de plusieurs clauses. C'est le mécanisme d'inférence qui dirige les actions à exécuter pendant la consultation. Exemple:

```
RULE 15
IF act = VUE_APPAREIL
THEN action = pince
FIND pince;
ASK pince: "La pince du robot est libre?"
CHOICES pince: Oui, Non;
```

Le bloc RULES qui suit le bloc ACTIONS contient la base de règles du système expert. Il représente la partie du système contenant toute "l'expertise" de l'expert. Exemple de règle:

```
IF nbr_grand = 0 AND nbr_moyenne = 4
THEN act = PREND_APP;
```

Le bloc STATEMENTS contient les instructions: ASK, CHOICES et PLURAL. L'instruction ASK crée un message qui demande à l'usager l'introduction d'une valeur pour une variable donnée. L'instruction CHOICES permet la création d'un menu avec plusieurs options. Elle accompagne toujours l'instruction ASK pour permettre à l'usager le choix d'une valeur pour la variable. L'instruction PLURAL identifie les variables pour lesquelles on assigne plusieurs valeurs durant la consultation.

7. Fonctionnement du prototype

En ce qui concerne le prototype, les informations normalement fournies par les capteurs sont fournies par un utilisateur, qui joue le rôle du module de perception. Le prototype génère le plan d'actions exécuté par le robot. Donc actuellement le sytème travaille sur la base d'un dialogue avec l'utilisateur.

Dans la version opérationnelle, le "dialogue" s'etabliera entre le système expert et le module de perception du système robotisé dans le but de générer le plan d'actions du robot. Le système global devient ainsi un système robotisé à base de connaissances.

8. Extension vers le système réel

VP-EXPERT permet la construction d'un prototype du système expert. Son mode de raisonnement est très proche de celui du PROLOG; tout d'abord par sa faculté de raisonner sous forme de faits et de règles, selon la philosophie de la logique formelle. De plus, certaines de ses commandes rappèllent les points forts de PROLOG. L'instruction PLURAL, par exemple, définit très bien le rôle du "backtracking" (retour en chemin inverse, ou rétroactivité) qui permettra sous PROLOG de générer sans difficulté majeure tous scénarios possible et surtout à l'aide de méta-règles et d'heuristiques de recherche de plans, de choisir rapidement la meilleure décision.

Le coeur du raisonnement de PROLOG, l'unification, s'apparente à la manière dont VP-EXPERT gère ses ACTIONS.

Le glissement vers PROLOG se fait donc en douceur avec VP-EXPERT, permettant ainsi de conserver les solutions efficaces testées par le prototype, et transférées, de manière homogène, vers un système expert écrit en PROLOG.

9. Conclusion

Le développement du SEPT nous a permis d'examiner une voie originale de résolution pour le problème de génération de plans dans le domaine de la robotique. Cette voie propose l'utilisation de techniques d'Intelligence Artificielle, à travers un système à base de connaissance.

En fait notre ambition est d'étendre les concepts et les résultats que nous formaliserons vers un système expert paramétrable, de la classe d'automates indépendants.

Les automates indépendants sont les systèmes robotisés dont les décisions (issues de son propre système à base de connaissance) agissent sur le monde extérieur, par opposition aux systèmes dont les décisions (toujours issues de leur système à base de conaissance) agissent sur eux-même, comme les systèmes de diagnostique par exemple.

La généralisation de nos résultats se fera cepandant dans des cadres déterminés en fonction du domaine d'exploitation, et prendront la forme d'une coquille. Notre système sera donc paramétrable, mais nous respecterons toute dépendance en foction du domaine d'application ou de spécialisation, pour éviter d'éventuels dérapages sémantiques que pourrait générer le système lors de l'autointerprétation de son environnement, et de ses conclusions.

Cette généralisation est envisagée sous forme codée de méta-règles, prudemment établies en vue d'un paramétrage étendu du système expert.

10. Références

1. R.Freedman, "Choosing an Expert System Shell/ Evaluating Shells", A.I. EXPERT - *The Magazine for the Artificial Intelligence Community* vol.2,no.9 september 1987.
2. S. Aguiar, "Inside VP-EXPERT" *PC-AI-The Artificial Intelligence Journal for Personal Computing* Summer 1987.
3. Tools: VP-EXPERT, *Expert Systems Strategies*, vol.3, no.3, march 1987.
4. "VP-EXPERT Rule Based Expert System Develop ment Tool", Paperback Software International, 1987
5. A. Bonnet, J.P. Haton, J.M. Truong-Ngoc, "Systèmes experts vers la maîtrise technique", InterEditions (1987).
6. C. H. Dominé "Techniques de l'intelligence artificielle", Dunod informatique, 1989.
7. J.N. Chatain, A. Dussauchoy, "Système Experts méthodes et outils", Eyrolles, 1987.
8. H. Farrany, "Les systèmes experts - principes et exemples", Cepadues Editions, 1987.
9. E. Rich, "Artificial Intelligence", McGraw-Hill Book Company, 1986.
10. J. L. Laurière, "Intelligence Artificielle - représentation des connaissances", Eyrolles, 1988.
11. J. M. Karkan, G. Tjoen "Systèms Experts - un nouvel outil pour l'aide à la décision", Masson, 1993.
12. D. Waterman, "A Guide to Expert Systems", Addison-Wesley Publishing Company (1988).
13. F.Hayes-Roth, "Building Expert Systems", Addison Wesley (1986)

90SC20

Functional Abstraction for High-level On Line Robot Programming

Colin Archibald[*]
Autonomous Systems Laboratory, Institute for Information Technology
National Research Council of Canada, Ottawa, Ont., Canada K1A 0R6

Emil Petriu[†]
Department of Electrical Engineering, University of Ottawa
Ottawa, Ont., Canada K1N 6N5

#149677

Abstract

Functional abstraction in robot programming is the explicit representation of robot operations in decreasing detail. At the highest level of abstraction the programmer is able to specify the robot program in terms of the physical requirements of the application. These physical actions are referred to as skills. *Robot skills are any action that the robot is capable of executing consistently. Libraries of skills, from simple motions to complex sensor-based skills, are made available to a robot programmer. Each skill is represented by an icon, which is parameterized to modify the behaviour. Robot operations are created on line without textual programming by linking icons in sequence and using simple branching constructs.*

Introduction

The evolution of industrial robot programming has been influenced by the increased capacity of microprocessors and the development of a variety of new sensing devices. For robots that do not use sensors to adapt to the habitat, teaching by showing and point-to-point teaching are still viable programming methods. However, sensor-based robots which adapt to the conditions in the environment require some form of intelligence which effects an appropriate action from raw sensor data. There are two methods of approaching this problem. The most common is to attempt to build a robot control system that has planning capabilities built into the control. The objective is to automatically generate a set of robot motions that will physically achieve a goal state which is described mathematically. This has been referred to as *task level* robot programming [1, 2]. The alternative to creating systems that plan the operation from a set of objectives is to create an interface between the human programmer and the robot that facilitates explicit specification of the operation.

Teach pendants have been the most common interface between the shop floor robot programmer and the robot for explicit robot programming. Although the functionality of teach pendants varies, they have generally been used for moving the robot to specific locations that are then taught to the controller. A playback mechanism allows the robot to iteratively follow the path through these locations. As the requirements of robotic tasks increased in complexity, teach pendants also became more complex, to the point where these interface devices may now contain full QWERTY keyboards and an additional 50 function keys. The increasing complexity of this human-robot interface is not in keeping with the progress in the field of human-computer communication. Shop floor programming is exceeding what can be reasonably expected of the shop floor programmer for reasons of complexity alone.

It is necessary to redefine the requirements of the shop floor programmer (hereinafter called the *applications specialist*) and to divide the responsibilities of creating a robot operation between the applications specialist and the *systems specialist*. Each of these individuals or teams has a very different mental model of the requirements for a robot program. An effective programming paradigm will permit them to work within their respective mental models of the problem.

A new approach to robot programming has been developed that provides a non-textual programming environment which allows the applications specialist to express the robot operations in terms of a sequence of robot skills. In addition to the programming environment, an architecture for the creation of these robot skills, which can take advantage of the capabilities of recent sensor and microprocessor technologies, has been developed. The resulting computational paradigm is called SKORP - SKills-Oriented Robot Programming. The important issues which arise in making such a paradigm practicable involve the tradeoffs between capability, flexibility, and usability at the non-textual programming level and the consistent interface between the non-textual shop floor programming and the systems programming.

---

[*] e-mail: archibald@iit.nrc.ca

[†] e-mail: petriu@trix.genie.uottawa.ca

The case for distinguishing between the applications specialist and systems specialist is found in the next section. An implementation of an icon-based shop floor programming environment is presented, along with an architecture designed specifically for the creation of the robot skills that these icons represent.

Division of Responsibilities in Robot Programming

One of the most important issues in robot programming, as in human-machine communication in general, is the level of abstraction in the interaction. A communication mechanism (usually a formal language) must satisfy both epistemological and practical requirements. The robot programmer must be able to express all of the required concepts in an unambiguous way, using tools that provide a means of meeting hard realtime deadlines, physical (on line) debugging, and rapid prototyping. If the robot programmer is responsible for every detail at all levels, the programming effort will be unmanageable. This issue is addressed in SKORP by separating the responsibilities of the developers into two levels. This separation is imperative considering the diverse skills required of the applications specialist and the systems specialist. The objective is to reduce the spectrum of knowledge required by individual members of the development team.

The attention of the applications specialist must be directed toward the application and not the programming details. In SKORP the applications specialist works only with icons, menus, and dialog boxes; i.e., no textual code is created at this level. The systems programmer uses a layered computational architecture to create the skills and make them available to the applications specialist via icons and dialog boxes. This division of responsibilities also provides a mechanism for robot software accretion and reusability.

Iconic Robot Programming

The objective of the iconic interface is to give the applications specialist as much information as is necessary at any particular time without encumbering him/her with detail that is not at the correct level of abstraction for the task at hand. As mentioned, each skill is represented by an icon which can be dragged around the screen. The icons contain visual cues, including colour and shape, to make the recognition of each one easier. The icons are functionally mutually exclusive. Only one skill icon is executing at any particular time. Execution order of the icons is intuitive. Each icon has one or more input connectors, which are at the left of the icon, and one exit connection, on the right of the icon. For icons that have multiple input connectors, each of the input connectors is functionally equivalent. There are a few special icons that do not represent skills. They are used to indicate the execution order of the skill icons. A START icon has no input connector. A CASE icon has a single input connector and multiple outputs and a MERGE icon has multiple inputs and a single output.

The progression of the execution of the operation can be thought of as passing a single execution token from icon to icon along the connector lines. There are only three control constructs: sequence, case, and iterate. The sequence construct is generally from left to right in the window and always from an input connector on the left of the icon to the output connector on the right of the icon. When the right side of the window is encountered the programmer can either "wrap around", by connecting the output connector of an icon at the right of the window to the input connector on the left of the screen, or scroll the window so that some of the icons are not visible. Any conditional execution of icons is accomplished using the CASE icon. This icon receives a datum from the previous icon and, based on this, passes the execution token to any one of its output connectors. The iteration construct is explicit in the network of icons that is constructed. If an output connector is attached to an input connector of an icon that has already been executed, a repeat loop exists. Nested repeat loops are also valid. Exits from loops are accomplished using the CASE icon. Note that many industrial robot tasks have no exit. The objective is often to define an infinitely repeating operation.

Iconic Programming Examples

A few examples of how the applications specialist's interface might look will make this description clear. Suppose the shop floor programmer wishes to describe a pick and place operation using skills. Fig. 1 shows the iconic representation of a simple pick and place operation. The START icon does some initialization of sensors and controllers and provides a starting icon for the operation. Three skill icons are shown: Move To Location, Pick, and Place. The control is from left to right except where the Place skill exit wraps around to the

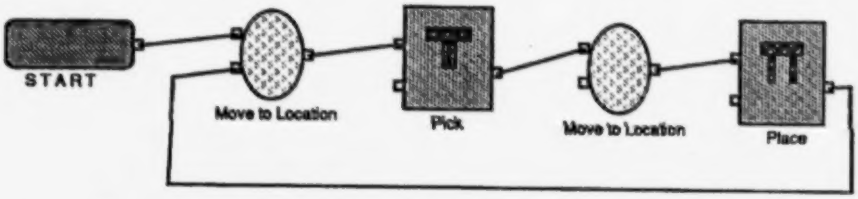

START
Move to Location
Pick
Move to Location
Place

Figure 1: An iconic description of a simple pick and place operation.

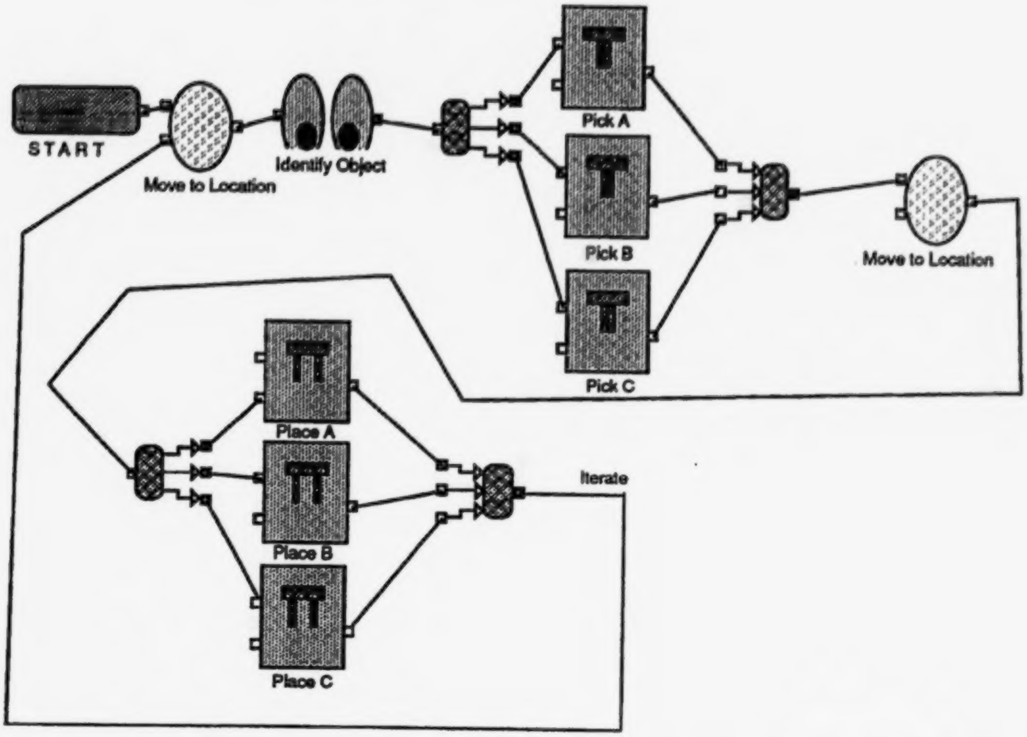

START
Move to Location
Identify Object
Pick A
Pick B
Pick C
Move to Location
Place A
Place B
Place C
Iterate

Figure 2: A pick and place operation that uses a sensor to identify the object and does a selective pick and a selective place.

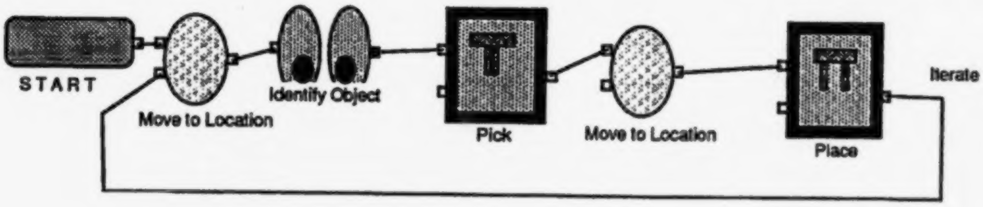

START
Move to Location
Identify Object
Pick
Move to Location
Place
Iterate

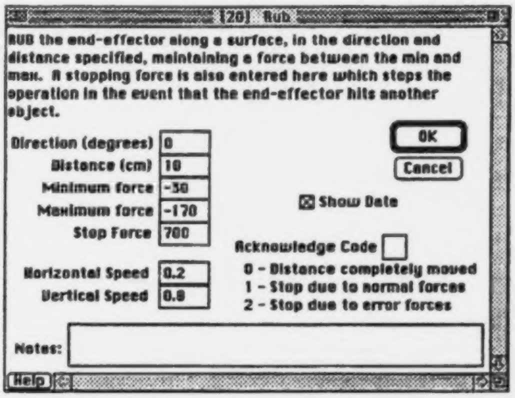
Figure 3: Pick and place using a sensor as in Fig. 2, with the selective pick and selective place portions collapsed into a complex icon.

initial move, creating a repeat loop. There is no exit, except for a STOP button at the bottom of the window (not shown). In Fig. 2 a slightly more complicated scenario is depicted. Suppose the environment consists of some transport mechanism that will deliver any one of three parts. The task is to identify which of the objects is arriving and to perform a Pick skill and a Place skill specific to that object. As can be seen in the figure, the programming window can become cluttered. In the figure, the execution path moves from right to left after the second "Move to Location" skill. Although it is easier to produce a figure on paper by doing this, the robot programmer will find it simpler to scroll the window so that only a portion of the operation is visible at any time. Another option for producing an entire operation in a single window is to further abstract the operation by combining a group of related skills into a single, more complex icon. For example, in Fig. 2 the Pick skills and the Case and Merge icons could be easily abstracted into a single icon and, similarly, the Place skills into another icon. In Fig. 3 this abstraction has been performed. The icons that represent an abstraction of other icons are indicated by a heavy outline and are obvious to the programmer who can pop these open to reveal the underlying icons at any time.

Popping Open an Icon

Each skill icon can be popped open to present a dialog window. This customized window offers the applications specialist all the options for this skill. Default values appear in the dialog box to give the programmer reasonable starting parameters. A Help button in this window opens another window, which contains a full textual description of the skill including preconditions and post conditions to help the programmer decide when to invoke the particular skill. The objective here is to make all the necessary information available, but not to encumber the programmer by presenting more information than is immediately necessary.

Besides the Help button, which describes the skill textually, there can be other buttons that provide more information in textual or graphical form. Fig. 4 shows an example of the dialog window for a *Rub* skill. This skill uses a wrist-mounted force-torque sensor to move the robot end effector along a surface while maintaining pressure within specified bounds. The dialog box contains several options for modifying the behaviour of the rub. Modifying the behaviour of *Rub* in terms of speed and pressure provides a powerful and flexible way that a skill can be reused for a variety of applications.



[20] Rub

RUB the end-effector along a surface, in the direction and distance specified, maintaining a force between the min and max. A stopping force is also entered here which stops the operation in the event that the end-effector hits another object.

Direction (degrees) 0
Distance (cm) 10
Minimum force -30
Maximum force -170
Stop Force 700
Horizontal Speed 0.2
Vertical Speed 0.8

☒ Show Data

Acknowledge Code ☐
0 - Distance completely moved
1 - Stop due to normal forces
2 - Stop due to error forces

Notes:

Figure 4: A sample of a dialog window for the *Rub* skill.

Implications of a Single Execution Token

The execution of a SKORP program progresses as a single execution token is passed from one icon to the next. The token branches based on the values of global variables which are returned from previous skills and stored on a *note pad*. There can be no parallel paths of execution since there is a single token. Multiple tokens are being considered for multi-robot workcells where each robot plays an equally important role. For workcells where there are multiple machines subservient to one robot, a single execution token is most appropriate.

The implication of a single execution token in terms of a formal language is that there is no conjunction operator. It is not possible to construct a network where the order of operation depends on multiple parallel preconditions. For example, it is not possible to express: *If A is completed AND B is completed, then perform C.* This capability has been carefully considered and, although it is occasionally desirable to create such a construct, it is almost never a necessity. For example, it is almost always possible to perform this operation in sequence, i.e., Perform A, Perform B, then Perform C. When a conjunction is essential, this must be expressed within a single skill, which effectively passes any task with this type of complexity to the systems programmer who is responsible for the creation of the skills.

The most important implication of the single execution token is that the functionality of the operation is obvious in a glance at the iconic description. As the operation executes, the skill that is currently being invoked is highlighted graphically. Since there is only one execution token, it is very easy to watch the execution of the operation on the iconic window and immediately isolate any errors or fine tuning requirement. This debugging method is feasible because complexity was intentionally omitted from the flow of execution design.

A Layered Robot Programming Architecture

High-level on line robot programming is possible only if there is some underlying computer architecture capable of executing the skills and communicating the results of the skills back to the application programming environment. SKORP is designed to have two computing systems: a workstation and a realtime subsystem. The application programs are created on the workstation using the iconic programming as described. This machine is a general purpose computer such as a PC or Macintosh, which is not designed for realtime programming but does have tools for human-machine interaction. The realtime subsystem is likely to have a very different hardware architecture from the workstation. It is clear that multiple processors in some configuration are necessary for multi-sensor multi-robot control [3] and that these multiprocessors must be designed specifically for hard realtime[1] programming.

Figs. 5 and 6 depict the layered architecture of SKORP. The emphasis is on programming abstraction, which is the removal of detail from those parts of the software where it is not required. Beginning at the top of Fig. 5, the *Operation* is the most abstract description. Operations consist of some number of specific skills, which are represented by icons. A specific skill is a parameterized version of a generic skill, e.g., the *Rub* skill can be parameterized into a *Polish* skill for a particular application. The arrow at the right of the figure indicates that the applications programmer develops the operation top down from a description of the operation to a collection of specific skills that meet the requirements.

Communication between the workstation and the realtime subsystem is through a realtime dispatcher module located on the realtime subsystem computer. The systems programmer is required to create all the modules below and including the realtime dispatcher. The arrow pointing up indicates that the systems programmer creates modules in a bottom-up order. The objective of creating a generic skill is met by first creating (or more likely reusing) the modules called machine controller, sensor driver, and logical sensor. These modules are created as objects in the object-oriented programming sense. They are encapsulated; other objects cannot modify the data within these objects

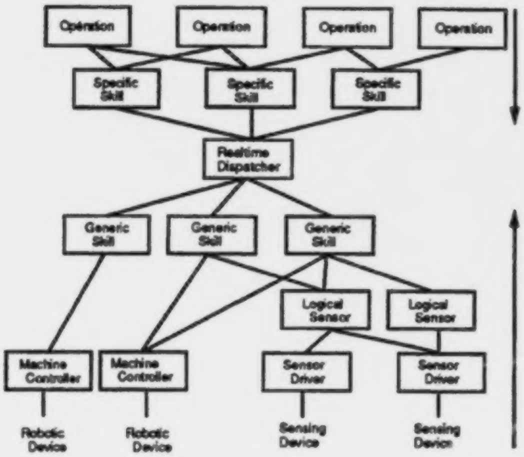

Operation
Operation
Operation
Operation
Specific Skill
Specific Skill
Specific Skill
Realtime Dispatcher
Generic Skill
Generic Skill
Generic Skill
Logical Sensor
Logical Sensor
Machine Controller
Machine Controller
Sensor Driver
Sensor Driver
Robotic Device
Robotic Device
Sensing Device
Sensing Device

Figure 5: The layered architecture for SKORP.

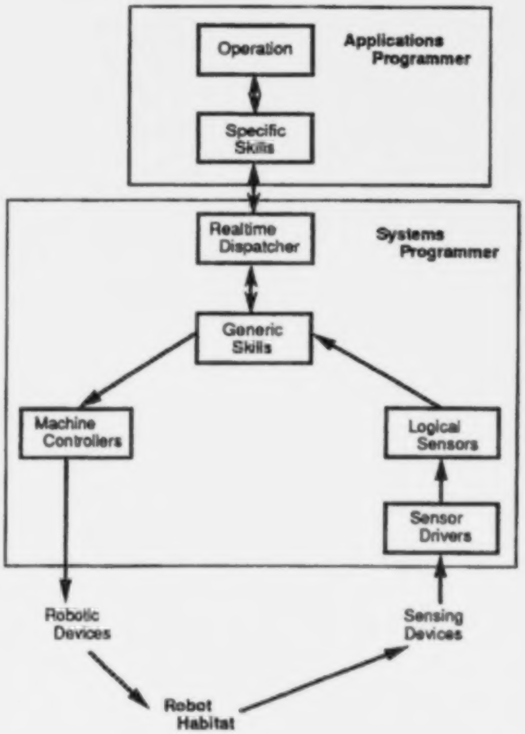

Operation
Applications Programmer
Specific Skills
Realtime Dispatcher
Systems Programmer
Generic Skills
Machine Controllers
Logical Sensors
Sensor Drivers
Robotic Devices
Sensing Devices
Robot Habitat

Figure 6: Information flow through the layered architecture.

---

[1] "Hard realtime" refers to systems where meeting absolute time deadlines must be guaranteed.

except through message passing. Each object has a data structure defined specifically for communication with that object. Once the object is created, any other object on an adjacent layer can communicate with that object by sending an appropriate message.

The information flow through SKORP is shown in Fig. 6. The most time-critical information flow is the sensor feedback loop indicated between the generic skill and the robot habitat. For each skill, the information communicated from the realtime subsystem to the workstation must be carefully designed such that enough information is available to the applications specialist to create a specific skill, but not so much as to impede the realtime performance of the entire system.

Experimental Implementation

The current prototype implementation of SKORP uses a Macintosh for the workstation, and a Harmony[2] operating system [4, 5] with four 68020 processors executing in parallel. Several skills have been created; sometimes the individual skills have become interesting research projects in themselves [6]. A laser range finder and a force-torque sensor have been used to create skills to rub a surface, follow an edge in six degrees of freedom, align the end effector normal to a surface, pick, place, palletize, etc. In many cases it was found that using an iconic interface is useful not only for the applications specialist, but also for the systems specialist, who can quickly create a dialog box to allow access to parameters that are useful for debugging, although not appropriate for use by applications programmers. For example, the systems specialist may want to vary coefficients to a control algorithm which the applications specialist will not be able to access. After the generic skill is functioning correctly, the dialog box is redesigned to provide access to the the skill's parameters that can be modified by the applications specialist.

Conclusions and Future Work

A prototype of an iconic programming environment has been created specifically for use by applications specialists who are not computer scientists. The realtime subsystem is based on a layered software architecture using object-oriented methods. Several skills using wrist-mounted sensors have been created and demonstrated in this environment. An interface that presents only the information required at any particular time is more usable than an interface based on a formal textual language. Combining existing skills and modifying their behaviour via dialog boxes has made the demonstration of new skills in the robotics laboratory considerably easier, and it is expected that industrial applications of robots using sensors can also be implemented more efficiently.

The Macintosh-Harmony-PUMA prototype is functioning as a demonstration of concept. An implementation on a PC using MS-Windows is currently being developed. The possibilities of using this formalism for other purposes, where the application specialist is distinctly different from the systems specialist, are also being explored. Programmable machines other than robots are being investigated, particularly lathes and milling machines. Fields where it is reasonable to have application specialists using systems in which they do not fully understand the systems level are under consideration. The first of these being investigated is a system for image processing applications.

Acknowledgements

The authors would like to thank E. Stuck and E. Kidd for their useful comments on an earlier version of this manuscript.

References

[1] G. S. Guler. Task level robot programming. *Proc. of 3rd Int. Sym. on Computer and Information Sciences*, pages 621–630, Commack, NY, Nov. 1988. Nova Science Publishers.

[2] G. D. Hager. *Task-directed Sensor Fusion and Planning – A Computational Approach.* Kluwer Academic Publishers, 1990.

[3] E. Solbach and A. Goldenberg. Real-time control of robots: Strategies for hardware and software development. *J. of Robotics & Computer-Integrated Manufacturing*, 6(4): 323–329, 1989.

[4] W. M. Gentleman. Using the Harmony operating system. ERB-966, National Research Council of Canada, March 1987.

[5] W. M. Gentleman, M. Wein, S. A. MacKay, D. A. Stewart, R. Parr, and D. Green. Harmony: An operating system for embedded industrial multiprocessor applications. *Proc. of IEEE Montech - COMPINT '87*, Montreal, November 1987.

[6] C. Archibald and E. Petriu. Robotic skills development using a wrist-mounted laser range finder. *Proc. of IEEE Instrumentation/Measurement Technology Conf.*, pages 448–452, Irvine, CA, May 1993.

---

[2] Harmony is a mark reserved for the exclusive use of Her Majesty the Queen in right of Canada by Canadian Patents and Development Ltd./ Societe canadienne des brevets et d'exploitation.

90C215

Knowledge Based Task Planning for Supervisory Control of Telerobotic Systems

Patrick Feighan, Keith Chrystall, Aldo Dagnino
Alberta Research Council, Advanced Computing and Engineering
6815 - 8 Street NE, Calgary, Alberta T2E 7H7
ph: (403) 297-2600, fax: (403) 297-2339
(feighan, kgc, dagnino)@skyler.arc.ab.ca

#149678

Abstract

This paper is based on supervisory control research completed under the Canadian Space Agency (CSA) Strategic Technologies in Automation and Robotics (STEAR) program. A task planning algorithm for supervisory control of space-based and terrestrial applications has been developed and is explained. The use of a knowledge base to provide world modeling capabilities for representing variable environments is also discussed. A scenario is presented for supervisory control of a remote telerobotic system, in the presence of time delay, where the results of task decomposition are verified by a graphic simulation of the activities planned. An operational proof-of-concept knowledge based task planning system which supports this control scenario is described. In addition to presenting the current state of development of the knowledge based task planning system, the paper also includes a discussion of future research directions which include integration with a trajectory planning system and the development of error recovery capabilities. System technologies are discussed for application to supervisory control of telerobotic systems for field applications.

1. Introduction

The Mobile Servicing System (MSS) is an advanced telerobotic system under development for the Canadian Space Agency (CSA) as part of Canada's contribution to the International Space Station program. The MSS will assist in assembling, servicing, and maintaining the Space Station. The CSA created the Strategic Technologies in Automation and Robotics (STEAR) program to support the development of advanced capabilities beyond those provided by the baseline MSS design. The Alberta Research Council (ARC) is participating in several projects under the STEAR program. This paper is based specifically on research and development in the area of supervisory control.

Supervisory control differs from direct manual control (teleoperation) in that the human operator acts as a supervisor of actions performed autonomously by the telerobotic system as opposed to directly providing all control inputs. Supervisory control technology augments the capabilities of human operators rather than attempting to remove humans entirely from the control loop. Supervisory control systems combine the best capabilities of humans, computers and automated equipment to achieve peak performance within the confines of current technology.

Applied to the MSS, supervisory control will reduce astronaut workload in the performance of maintenance tasks and support the development of effective ground control of the MSS. These same supervisory control concepts are applicable to a wide variety of terrestrial telerobotic systems, such as unmanned vehicles for mining and hazardous material handling. In addition, concepts from task planning, which is a crucial element of supervisory control as discussed in detail in this paper, are applicable to the programming of automation systems for manufacturing applications.

2. Supervisory Control Scenario

In a supervisory control application, the human operator acts as a supervisor to one or more telerobotic systems which autonomously carry out planned activities. The duties of the human operator include the generation and verification of these planned activities, monitoring the activities of the telerobotic system , and providing manual assistance or override in emergencies or situations that the telerobotic system cannot handle autonomously.

Supervisory control can improve the efficiency of human operators as compared to direct teleoperation. The human operator takes on a supervisory role instead of having to provide every input to the telerobotic system manually. Inputs from a supervisor can take the form of a few high level commands instead of many detailed numeric or manual inputs.

Supervisory control can also facilitate control of telerobotic systems for applications where teleoperation is impractical. Such applications may

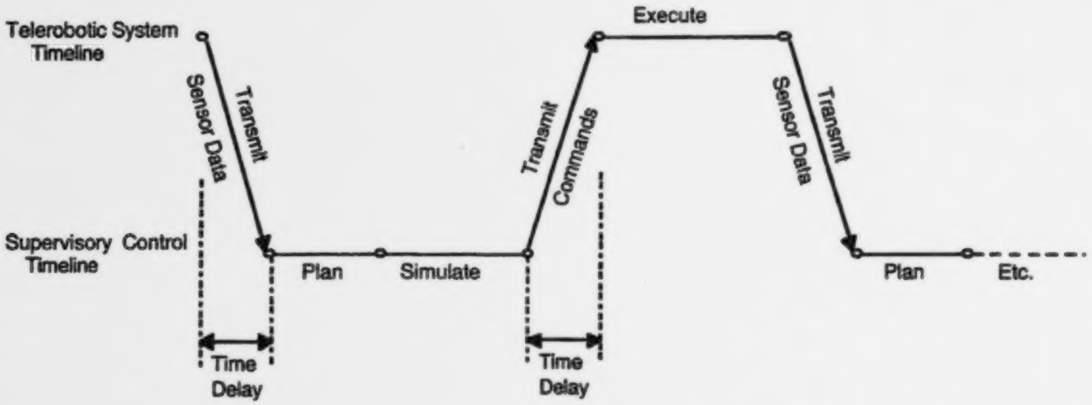


FIGURE 1: Supervisory Control Scenario for MSS Research

involve remote operation of a telerobotic system in the presence of time delays or communication restrictions. Other applications that preclude teleoperation are those requiring a single operator to control multiple vehicles.

Enabling an operator to control a telerobotic system by means of high level, non-real time inputs requires the automatic generation of lower level plans in an executable form. An automatic task planning element provides this function. Task planning automates the breakdown of high level commands that are input by the human operator into sequences of low level instructions that may be executed autonomously by the telerobotic system.

The successful generation of task plans is based on the availability of accurate information representing the current environment of the telerobotic system. One mechanism for representing this data is a computer knowledge base. Due to the variability in the environment of telerobotic systems, such a knowledge base requires the capacity for updates based on data received from sensors in the vicinity of, and onboard, the telerobotic system.

To support the human operator's role of verifying task plans before they are executed by the telerobotic system, a useful feature of the supervisory control system is a predictive simulation capability. Predictive simulation is a computer generated estimate of what the telerobotic system will do when it executes a task plan. Results may be displayed to the operator as a graphical animation of the telerobotic system carrying out the planned activities.

In addition to the features that the control system must have, supervisory control is only feasible if the telerobotic system also has certain features. The telerobotic system must be able to execute low level commands autonomously, return sensor and system status data to the control system, and to protect or 'safe' itself in the event of an error or unexpected condition.

Based on all of the features described above, the supervisory control scenario that was developed for the MSS is illustrated in Figure 1. This application is one in which the telerobotic system aboard Space Station must be remotely operated from the ground. Communication time delays of 3-10 seconds are expected.

The scenario begins with the supervisory control system receiving current sensor data from the telerobotic system. This data is used to update the representation of the telerobotic system environment in the world model knowledge base. Following this, the human operator enters high level commands which describe the activity that the telerobotic system is to perform. These high level commands are typically developed during off-line development and simulation of necessary tasks.

The task planning system, utilizing information in the world model knowledge base, then decomposes the high level commands into a sequence of low level instructions that are to be executed autonomously by the telerobotic system. The sequence of executable instructions is generated up to the point where additional sensor information is needed (e.g. vision system location of an object) for the next step of the overall task to proceed. The execution of this sequence is simulated and displayed to the operator for verification.

Once approved, the sequence of instructions is transmitted to the telerobotic system for execution.

When execution is complete, new sensor data is sent to the supervisory control system. The cycle continues as additional plan segments are generated, previewed, and executed until the activity is completed. Efficiency is increased as the duration of planned activities between necessary sensory updates is increased.

3. Task Planning System Description

A knowledge based task planning system was developed by the ARC to support supervisory control of an MSS telerobotics research testbed. The telerobotics testbed includes a manipulator and tooling that emulates a subset of anticipated MSS functionality. A vision system recognizes pre-defined objects (by virtue of a special symbol placed on an object's surface), and can determine object position and orientation. The telerobotics testbed has low level controllers which can execute low level instructions with the testbed devices.

The architecture of the task planning system (Figure 2) utilizes several concepts derived from the NASREM architecture [1]. NASREM is an acronym for the NASA/NBS Standard Reference Model for Telerobot Control System Architecture. The NASREM concepts which were adopted include the heterarchical division between world modeling and task decomposition systems, and the hierarchical division of task decomposition into discreet layers of detail. The world modeling system maintains a computer representation of the environment of the telerobotic testbed. The task decomposition system decomposes high level commands into low level instructions for the telerobotic testbed to execute autonomously. An operator interface allows the human operator to make inputs and inquiries and to intervene at any level of the architecture.

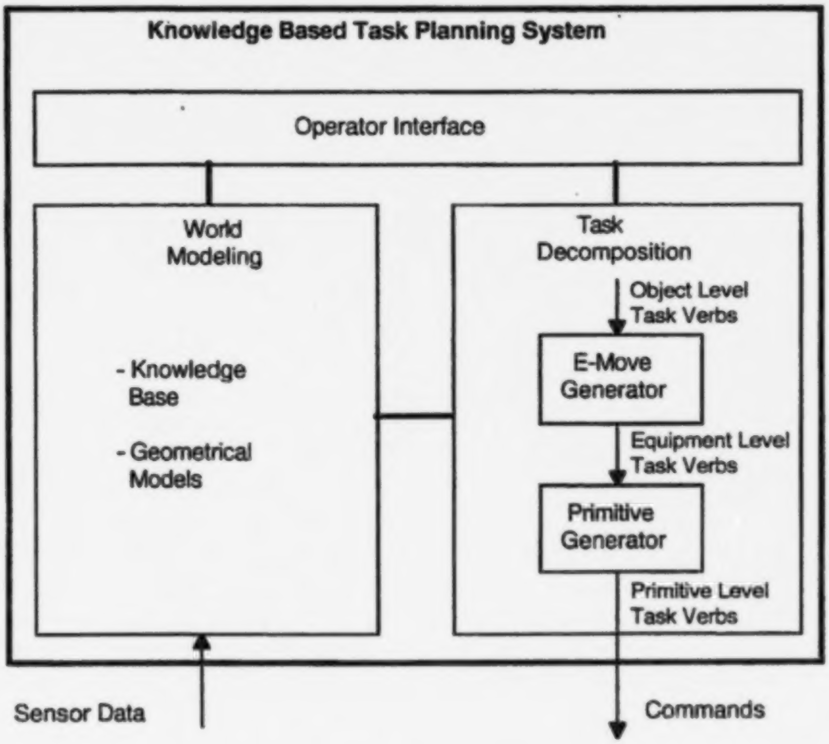

Knowledge Based Task Planning System
Operator Interface
World Modeling
- Knowledge Base
- Geometrical Models
Task Decomposition
Object Level Task Verbs
E-Move Generator
Equipment Level Task Verbs
Primitive Generator
Primitive Level Task Verbs
Sensor Data
Commands

FIGURE 2: Control Architecture for Task Planning System

3.1 Task Decomposition System

Task decomposition is based on a set of defined task verbs that provide a framework allowing the activities of the telerobotic testbed to be specified and decomposed. Task verbs are defined for each of three levels: object, equipment, and primitive. Object level task verbs are the input commands provided by the human operator. They define actions to be performed on objects in the telerobotic testbed workspace, with no mention of which devices are to facilitate those actions. Equipment level task verbs symbolically assign specific devices to accomplish the actions defined by object level task verbs. Primitive level task verbs are low level instructions that are output to individual device controllers. They contain information specific to a device with all necessary numeric parameters.

The task decomposition system breaks down each task verb into its associated lower level task verbs when the necessary constraints and preconditions are met. A group of predicate operators provides contingency handling for a limited number of cases where constraints or preconditions for task verb decomposition are not met.

Task decomposition takes place in two steps, through the E-move generator and the primitive generator. The E-move generator decomposes the object level task verbs of an input script into equipment level task verbs. This converts object related actions into device-specific activities. The primitive generator decomposes equipment level task verbs into primitive level task verbs. This converts symbolic, device-specific activities into numeric output instructions that are executable by the device controllers of the telerobotic testbed. The two level hierarchy of task decomposition can be expanded upwards in the future to accommodate higher levels of planning.

The task decomposition system is developed in the object-oriented environment of ART-IM (Automated Reasoning Tool for Information Management). ART-IM is a product of Inference Corporation, El Segundo, California and provides a tool kit for the development of knowledge-based systems.

Task verbs were defined to describe maintenance operations that the MSS may be required to perform. These operations are referred to as ORU (Orbital Replacement Unit) changeouts. The task verbs defined for ORU changeout are listed below:

Object Level Task Verbs

ACQUIRE, DETACH
EXTRACT
PLACE, INSERT
RELEASE, ATTACH
OPEN, CLOSE

Equipment Level Task Verbs

manipulators:
ENGAGE, DISENGAGE
POSITION

tools:
GRAPPLE, UNGRAPPLE
FASTEN, UNFASTEN

Primitive Level Task Verbs

manipulators:
DE/ACTIVATE_OTCM
MOVE (straight line)
MOVE_H (sensor homing)
MOVE_J (joint vector)
MOVE_C (circular interpolated)

tools:
ACTUATE

vision systems:
CONFIGURE

Predicate operators were defined to provide contingency handling capabilities during the decomposition of ORU changeout task verbs. They can be invoked by the task decomposition system in cases where constraints or preconditions for task verb decomposition are not met.

Object Level Predicate Operators

CLEAR_OTCM
(remove a tool from an OTCM)
GET_TOOL
(acquire the correct tool)
STOW
(stow a manipulator)
UNSTOW
(unstow a manipulator)
LOCATE_ORU
(locate an ORU using vision)
LOCATE_SITE
(locate ORU/tool site using vision)
LOCATE_PANEL
(locate an access panel using vision)

In equipment level task verbs, manipulator positions are specified using symbolic poses. Each symbolic pose has an associated numeric breakdown that is utilized by the primitive generator to produce primitive level task verbs. Numeric pose calculations are performed using the matrix handling capabilities of CimStation, an interactive robotic simulation environment and product of SILMA Incorporated, Cupertino, California.

Symbolic poses are classified by pose type and pose class. Possible pose classes are:

- endpoint: specifying the manipulator to move to a desired endpoint
- homing: specifying the manipulator to move to a desired offset with respect to a vision system target
- joint vector: specifying the manipulator to move to a desired joint vector

The following pose types can be dynamically created by the task decomposition system for ORU changeout:

- grapple: move to the grapple fixture of an object
- postextract: move to complete the extraction of an ORU
- preinsert: move to begin insertion of an ORU
- ungrapple: move to location where a grappled object will be released
- locate_oru: move to position the camera to locate an ORU
- locate_site: move to position the camera to locate a site
- locate_panel: move to position the camera to locate an access panel
- open: move to open an access panel
- close: move to close an access panel

Current capabilities of the primitive generator include rudimentary trajectory planning for depart and approach movements between symbolic poses.

During task decomposition, plans are generated in segments between which sensor updates to the world model are required. Activities such as the predicate operators LOCATE_ORU, LOCATE_SITE, and LOCATE_PANEL require sensor updates before planning can continue. Sensor updates are specified by ending an executable plan segment and adding a condition that requires sensor data to be read before decomposition proceeds. This is referred to as a monitor point.

3.2 World Modeling System

The world model provides the task decomposition system with information on the telerobotic testbed environment necessary for the evaluation of preconditions during the decomposition of task verbs. This information is represented in a knowledge base and through geometrical modeling. The knowledge base component is implemented in ART-IM and the geometrical modeling component is implemented in CimStation. The information in the world modeling system is kept current through updates from the task decomposition system (which generates post-conditions resulting from task decomposition), by updates from the sensors of the telerobotic testbed, and by manual updates from the human operator. Sensory updates are performed at monitor points which occur at system initialization and after the execution of each plan segment generated by the task decomposition system. Information from sensors includes manipulator joint angles and endpoints, manipulator tool status, and vision system data providing the locations of objects such as ORUs.

The following knowledge base object types are supported by the world modeling system:

ORU
Access Panel
Manipulator
Vision System
Tool
Tool Mechanism
Site
Pose

The geometrical models of the world modeling system provide three-dimensional geometrical representations of objects in the telerobotic testbed environment. These geometrical representations are used for the calculation of numeric poses, for the generation of synthetic graphical images for the operator interface, and for the determination of possible collisions during the simulation of plan segments. A set of geometrical models was created for each telerobotics testbed that was controlled. One such set is shown in Figure 3. The geometrical modeling component of the world modeling system is kept consistent with the information in the knowledge base, as it is the knowledge base that is kept current by the task decomposition system and the sensor data from the telerobotic testbed.

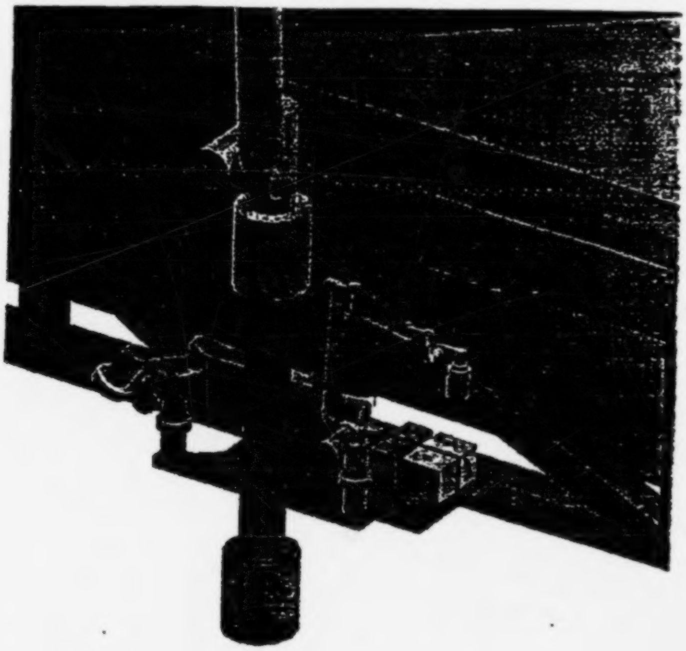


FIGURE 3: Geometrical Models for Mockup Space Station

3.3 Operator Interface

The operator interface provides access to all components of the knowledge-based task planning system. This supports supervisory control by a human operator monitoring remote operation of the telerobotic testbed. The operator interface provides the following functionality:

- initiate sensor update of world modeling system.
- display/edit information in world model knowledge base.
- create geometrical models.
- initialize orientation of geometrical models based on data in knowledge base.
- graphically display world modeling subsystem geometrical models.
- create, edit, and troubleshoot task decomposition rules.
- facilitate input to task decomposition system in the form of high level task verbs (ASCII files).
- initiate operation of task decomposition system.
- display/edit plan produced by task decomposition system.
- graphically display plan produced by task decomposition system as predictive simulation; indicate expected collisions.
- file management operations.

4. Task Planning System Operation

Operation of the task planning system involves the following steps:

Initialize world model: The operator queries the telerobotic testbed for current sensor and status data. When data is received, the task planning system stores the data with the appropriate knowledge base objects, performs any necessary coordinate system conversions, initializes geometrical models based on knowledge base data, and creates a backup of the knowledge base.

Verify world model: The task planning system displays knowledge base data and generates a graphical simulation using geometrical models. The operator compares simulation and actual video displays to assist in establishing the validity of the world model for plan generation.

Generate task plan segment: The operator enters a script of object level task verbs to be decomposed. The task planning system decomposes object level task verbs into primitive level task verbs for execution by the telerobotic testbed. Decomposition does not proceed if any of the necessary preconditions are not met or handled by predicate operators. Decomposition is halted when a monitor point is reached. The resulting sequence of primitive level task verbs is a task plan segment.

View predictive simulation of task plan segment: The task planning system displays a complete or step-by-step predictive simulation of the task plan segment. The operator approves the segment for execution, making manual edits if necessary.

Send task plan segment to telerobotic testbed for execution: The operator transmits the segment to the telerobotic testbed. The task planning system receives and logs sensor and status data as each primitive level task verb is completed.

Update world model: After execution of the last plan segment is completed, the task planning system finds the current sensor and status data in the data log. The world model is updated in the same manner as in the initialization described above.

Verify world model: In addition to displaying knowledge base data and a graphical simulation, the task planning system now compares sensed data to expected (modeled) data. The operator uses this information to establish that the previous plan segment was executed correctly and to establish the validity of the world model for continued plan generation.

Generate next task plan segment: The sequence of plan segment generation, verification, and execution is repeated until the input file has been completely decomposed and the task completed.

5. Conclusions

The knowledge based task planning system described in this paper was successfully tested through a series of project milestone demonstrations ending in June 1993. The accommodation of time delays proved to be an important feature. Task decomposition lessens the impact of time delay (as compared to teleoperation) by providing sequences of activities for the telerobotic system to perform. Interaction between the telerobotic system and operator occurs only at monitor points, rather than the continuous real time communication required by teleoperation techniques. Contingency handling was successfully handled by the predicate operators. These can be extended to a wide class of error and failure situations.

Future work is required to develop error recovery planning capabilities. The necessary features of task tracking, error representation, and replanning may be established within the existing system framework.

The current task planning system has only a limited developer interface. This should be expanded to better support the developer in editing and creating task verbs, predicate operators, task breakdown structures, and knowledge base objects. In addition, a complete operator interface is necessary to allow proper display and access to the knowledge base, and incremental display and editing capabilities with the simulator.

The task planning system needs to be 'ported' to a more generic platform to minimize dependence on specific commercial systems. There are several object oriented environments that would be satisfactory.

The task planning system performs only crude trajectory planning by establishing key poses and the necessary approach and depart moves. For use with the structured MSS environment, integration of the task planning system with a trajectory planning system is possible to automate the generation of collision-free trajectories. Currently, these must be added manually.

Knowledge based task planning can also be applied to supervisory control of telerobotic systems in unstructured environments such as those found in field (military, mining, and hazardous material handling) applications. Instead of the trajectory planning systems that can be used for structured environments, collision avoidance relies on sensing and control capabilities on-board the telerobotic system. Improved sensor technologies (e.g. detection of moving or stationary obstacles) are needed for more robust and reliable applications of field telerobotics.

In general, supervisory control provides an 'evolutionary' path for developing greater autonomy in telerobotic devices. Useful applications can be developed with existing technology. As advances are made in sensor, computer and information technologies these can adapted into supervisory control schemes permitting better performance on existing applications and allowing the development of new applications.

6. Acknowledgments

The work described in this paper has been completed by the Alberta Research Council for the Canadian Space Agency STEAR program under contract to International Submarine Engineering Ltd. of Port Coquitlam, BC, Canada and Thomson-CSF Systems Canada of Ottawa, ON, Canada. Project team members of these prime contractors and of their various subcontractors have made significant contributions to this work.

7. References

1. J. Albus, H. McCain, and R. Lumia, "NASA/NBS Standard Reference Model for Telerobot Control System Architecture (NASREM)", *National Bureau of Standards (NIST) Technical Note 1235*, July, 1985.

2. K. Chrystall, A. Dagnino, P. Feighan, "A Multi-Level Planning System for Robot Applications", *The Canadian Aeronautics and Space Journal*, Vol. 37, No. 3, pp. 139-143, September, 1991.

3. A. Dagnino, "A Hierarchical Planning Knowledge System Applied to Manufacturing Processes", *Handbook of Expert Systems in Manufacturing*, McGraw-Hill, Inc. publisher, R. Maux and J. Keyes editors, chapter 27, pp. 404-417, 1991.

4. E. Jackson, K. Buchan, D. Eddy, G. Springle, K. Chrystall, P. Feighan, and A. Dagnino, "A Testbed for the Development of Ground-Based Control of Space-Based Manipulators", *Proceedings of the Third Conference on Military Robot Applications*, Medicine Hat, Alberta, Canada, September 9-12, 1991.

5. P. Wojcik, P. Feighan, K. Chrystall, "Operator Interface Design for Space Telerobotics Under Supervisory Control", *Proceedings of the '91 International Conference on Intelligent Teleoperation*, Grensboro, NC, November 6-7, 1991.

6. P. Wojcik, P. Feighan, K. Chrystall, "Supervised Space Robot System - Operator Interface Design", *Proceedigns of the 42nd Congress of the International Astronautical Federation*, Montreal, Canada, October 5-11, 1991.

7. P. Feighan, A. Dagnino, K. Chrystall, "Knowledge-Based Task Planning for the Special Purpose Dextrous Manipulator", *Proceedigns of OE/Technology 92: Intelligent Robotics and Computer Vision*, Boston, Massachusetts, November 15-20, 1992.

924701

Design Knowledge Capture of Robotic Systems

Elan Yaniv & Walling R. Cyre

Virginia Tech
The Bradley Dept. of Electrical Engineering
Blacksburg, VA 24061-0111

#149679

Abstract

This paper discusses work being done on the development of a tool for automating the design knowledge capture of robotic systems. A major problem in the design cycle today is the lack of an automated system for capturing design knowledge from requirements and specifications that are expressed in informal notations, including natural language and diagrams. At Virginia Tech the ASPIN system is being developed to perform this task in the domain of digital microprocessor systems. The goal of the work reported here is to port the system to the robotics domain by developing an appropriate knowledge base.

The ASPIN system includes translators which map the various source languages (both textual and graphical) found in requirements and specifications into a common knowledge representational system. The knowledge representation selected is a form of semantic networks called conceptual graphs. Conceptual graphs derived from isolated requirements can be integrated using schemata. The integrated conceptual graphs may then be analyzed for consistency and completeness, and engineering models to support simulation, and cost/performance estimation can be constructed.

The knowledge base includes a catalog of concepts and relations with a concept type hierarchy, and a dictionary of the vocabulary of the source language with grammatical categories and with word meanings in terms of 'canonical' conceptual graphs. In addition, a library of schemata corresponding to word usages is needed to support integration of conceptual graphs. The steps in the development of the knowledge base for robots includes selection of a corpus of sentences on robots, derivation of the vocabulary, selection of concept types, the development of a type hierarchy, derivation of canonical conceptual graphs, and the development of schemata. The preliminary corpus supporting this development contains approximately 50 abstracts on robots, having a vocabulary over 1800 words.

Introduction

Automation is a key part of many different phases in the design cycle. Despite this, the interpretation of design specifications still remains a manual process, largely because they employ natural language and informal graphical notations. There is a definite need for a design automation tool that translates an initial description of a system into a form useful to design engineers for analysis and design synthesis.

This has been an active area of research at Virginia Tech in the domain of digital systems. The design knowledge capture tool, called the ASPIN system, currently under development uses a knowledge base to express descriptions of digital systems as a type of semantic networks called conceptual graphs. The ASPIN system is comprised of a knowledge base and programs to translate natural language descriptions and diagrams into conceptual graphs. The knowledge base contains the vocabulary and grammar used for describing digital systems, as well as a canonical basis of conceptual graphs which can be assembled to represent the meaning of sentences. By changing the information in the knowledge base, this tool can be ported to domains other than digital systems.

This project was Sponsored in part by Virginia's Space Grant Consortium Scholarship 069105, Semiconductor Research Corporation Contract 92-DJ-230, and NSF Grant MIP-9120620.

The work described in this paper deals with developing a knowledge base pertaining to robotics. This work also includes attempts to install this knowledge base with software previously developed to operate with the digital system knowledge base. In the following sections, conceptual graphs are introduced, followed by a brief description of the ASPIN system. Then, the derivation of the robotics knowledge base is described. Finally, the current implementation and plans for continuing the work are mentioned

Conceptual Graphs

Conceptual graphs [9] are a knowledge representation notation similar to semantic networks, but have been developed from a graphical form of symbolic logic called existential graphs [8]. A conceptual graph is finite, connected, labeled, and bipartite, and consists of two sets of nodes: concept nodes and relation nodes. Each concept and each relation is labeled by a type label, and the concepts may also have a referent label (or name) as well. Concepts may represent specific individuals (denoted by a # symbol in their referent field) or a generic concept (indicated by a *). In the graphical notation for a conceptual graph, concepts are represented by rectangles and relations, by ellipses. An example of a conceptual graph for Sentence 1 appears as Figure 1.

> *When the operator moves the hand controller, the robot's arm moves and joint angles are displayed on the user interface prototype"* (1)

Concept types are partially ordered by a subtype/supertype relation to form a concept type hierarchy, which is used in operations on conceptual graphs. Four basic operations (or formation rules) have been defined for conceptual graphs: copy, restrict, join, and simplify. Copy simply replicates a graph. The restrict operation replaces a type label by a subtype, or a generic concept may be replaced by an individual. The join operation allows two identical concepts (in the same or different graphs) to be merged into one by removing all arcs from one of the concepts and reattaching them to the other, and then deleting the isolated, duplicate concept. Finally, the simplify operation removes duplicate relations resulting from join operations.

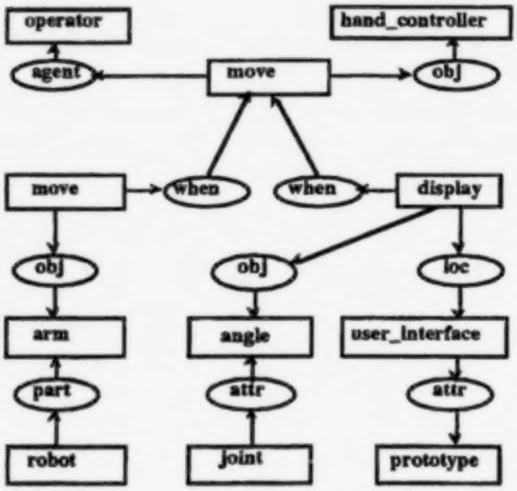


FIGURE 1: A conceptual graph

These four operations are used to assemble conceptual graphs such as the one in Figure 1 from a basic set of conceptual graphs (called canonical graphs) which represent the meanings of words in the vocabulary. Schemata, which are essential to integrating conceptual graphs from individual sentences, are conceptual graphs which represent typical usage of concepts, and tend to be substantially more complex than canonical graphs.

The ASPIN System for Digital System Design Synthesis

The ASPIN (Automatic SPecification INterpreter) system is being developed as illustrated in Figure 2 to capture specifications and requirements (design knowledge) expressed in a variety of notational systems including block diagrams, timing diagrams, flowcharts, and natural language. These expressions are then translated to a common knowledge representation system, conceptual graphs, in which they may be integrated. Diagrammatic requirements can be translated by relatively simple mapping algorithms, but even a restricted subset of natural language is highly ambiguous, and requires a more substantial process [1]. In the ASPIN system, a top-down, parallel, chart parser is used for syntactic analysis of English statements. The parse trees produced by the parser are then analyzed using the knowledge base of canonical conceptual graphs and a type hierarchy in order to construct conceptual graphs representing the meaning of sentences. Conceptual graphs representing sentences and diagrams must then be integrated using schemata [2].

The integrated knowledge derived from requirements may then serve as the basis for reasoning about the requirements, including analysis for inconsistencies, estimation of characteristics (cost and performance), and synthesis of designs in the form of engineering models [5]. The engineering models are expressed graphically and in a computer hardware description language called VHDL (VHSIC Hardware Description Language) [6]. The graphical model shown in Figure 2 is displayed to the specification author to validate the interpretation of English statements, which are generally ambiguous. At this time the parser and semantic analyzer have been developed and operate on a subset of the English sublanguage used in hardware descriptions of microprocessor products. A mapper has been constructed to translate diagrams represented as EDIF (Electronic Design Interchange Format) [10] netlists into conceptual graphs. Prototype programs to generate graphical and VHDL models of a limited range of conceptual graphs have also been developed. An algorithm has been designed to generate conceptual graphs from timing diagrams in terms of temporal logic relations. The system is being implemented in 'C' and Prolog in a UNIX environment with X-Windows.

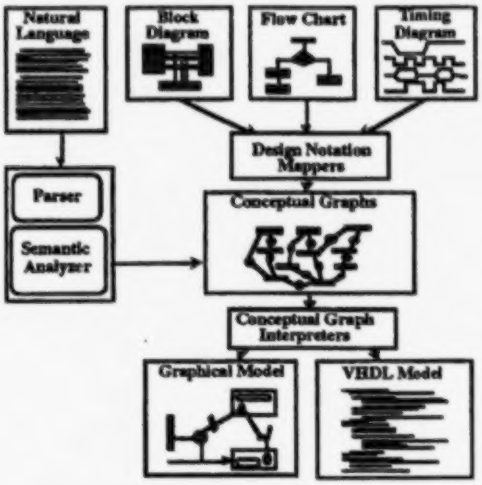


FIGURE 2: The ASPIN System

Knowledge Base Development for Robotics

The semantic domain in which the ASPIN system operates is determined by its knowledge base. Replacing the current knowledge base on microprocessors by a knowledge base on robots will allow the ASPIN system to reason about and synthesize engineering models of robots, with relatively little change in the mapping and inference software. A 3-dimensional geometrical drawing mapper will also be needed to accept physical descriptions of robots. The current diagram mapper is only adequate for 2-dimensional schematic illustrations. In developing the knowledge base, each diagram is considered to be written in its own type of graphical source language.

The development of the knowledge base for microprocessors involved several steps.

1) a corpus of sentences on the subject was compiled,

2) the vocabulary was determined (with word frequencies),

3) a phrase-structured grammar was developed to cover most of the language,

4) a subset of the vocabulary was selected as names of concept types,

5) the type hierarchy was determined,

6) the sentences were analyzed to determine word meanings for the vocabulary

A preliminary knowledge base for robots was developed in the project reported here by repeating all but step 3 above. (The requirements grammar for microprocessors was assumed to be satisfactory for robots as well [3].)

The first step in designing this knowledge base was to discover a vocabulary. This was automated with the use of a Pascal program that performs a lexical scan on the text inputted to it. Over 50 abstracts, consisting of 7,667 words, dealing with robotic systems, were given as input to the "Getwords" program. The "Getwords" program generated a vocabulary of 1,886 words from these abstracts. It also ordered the list by word frequency. The descending order list highlighted the words that are important in these types of descriptions. As an aside, the "Getwords" program also tracked the vocabulary growth as each abstract was inputted into the program.

The vocabulary list was an invaluable tool for focusing on a few of the more frequent words. A sampling of the words that appeared towards the top of the list are tabulated below with their word frequencies (singular and plural combined). Function words (articles, conjunctions and preposisitons) are not included in Table I.

Table I: Vocabulary Examples

systems	131	joint	21
robot	104	operator	18
space	67	position	18
manipulator	48	arm	12
motion	34	state	11
trajectory	32	hand	10
controller	27	camera	9
parameter	27	computer	9
task	27	end_effector	7
dynamic	22	inspection	2

The next step was to select the set of concepts and determine the type hierarchy. The selection began with the concepts: robot, camera and system. The hierarchical relationship among these was determined, and as more types were considered, they were placed appropriately in the hierarchy. At the top of the hierarchy shown in Figure 3 are concepts for device, action, event, person, and parameter which are immediate subtypes of the universal type, T (See Figure 3). For comparison, the top-level concepts in the microprocessor domain include object, action, event, state, and characteristic. Note that in this hierarchy, a manipulator is a system, whereas a link is a device, anticipating that the substructure and internal behavior of a manipulator is of interest, but that of a link is not. Type hierarchy information can also be obtained directly from the corpus of text by searching for instances of expressions of the form "X is a Y". In the sample corpus, these instances are repeated in Table II. The hierarchical knowledge gained this way must be used cautiously, because it may reflect what a domain expert would consider a transitive relationship. For example, the expression "an arm is a device that ..." suggests that arm ≤ device, but according to Figure 3, arm ≤ system ≤ device. These transitive relationships are used in analysis of sentences, but may be easily computed when required, rather than depending on error-prone manual derivation.

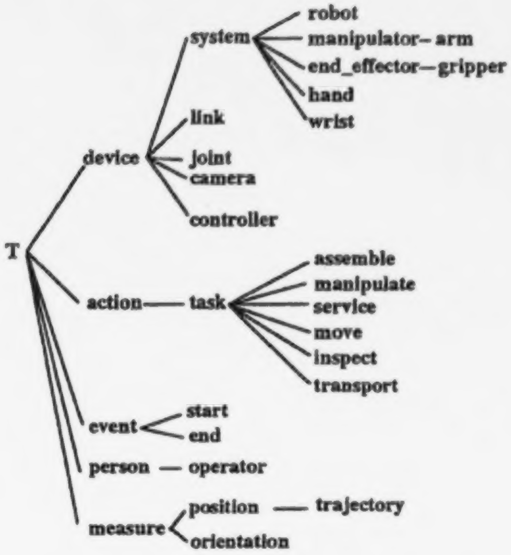


FIGURE 3: A Partial Type Hierarchy

Table II: "X is a Y" Instances

assemble is a task
controller is a device
manipulation is a task
orientation is a measure
position is a measure
service is a task
transport is a task
inspection is a task

Although not part of the dictionary, another body of information of interest is the structural hierarchy of assemblies and their components. This information is represented in the knowledge base as schemata in which the device concepts are related by 'has_part' relations. As with the type hierarchy, this can be developed manually by a subject domain expert or with general references [7], or can be derived in part by searching for expressions of the form "X has a Y". Simplified expressions of this form from the abstracts studied here are listed in Table III, and the schemata developed from them are illustrated in Figure 4, in which the has_part relation is abbreviated by a 'p'. Caution must be used since the verb 'has' can also indicate attribution as in the case of "load has mass." Also, "has a" expressions may reflect transitive has_part relations.

Table III: "X has a Y" Instances

system has camera
system has platform
robot has sensor
load has mass
manipulator has sensor
manipulator has controller

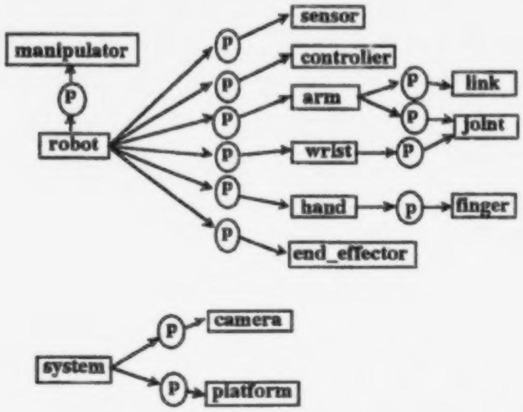


FIGURE 4: Has_part Schemata

The next step in developing the knowledge base is to develop the canonical conceptual graphs for the dictionary. For action and event concepts, this is performed for a particular concept by finding all sentences using the concept and decomposing them into a collection of case frames [4]. A case frame for a verb is a template having a set of slots called roles. For example, sentence (1) can be decomposed into the three case frames (2-4), where the first role is generally the agent of the action, the second item in a frame is the action concept itself, the third role is the object of the action, and subsequent roles are identified by their linking prepositions, as 'on' indicates a location role.

operator, move, hand (2)
arm, move (3)
something, display, angle, on interface (4)

By collecting a set of frames for each verb and generalizing the concepts used in the various roles, a canonical graph can be constructed for the dictionary.

In the robotics abstracts, the case frames for 'move' are listed in Table IV, and the canonical graphs derived from them are shown in Figure 5.

Table IV: Case Frames for 'move'

gripper, move, from position, to position
arm, move, to position
person, move, device
robot, move, gripper

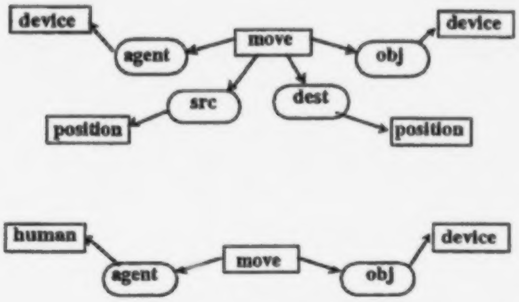


FIGURE 5: Canonical Graphs for "move"

It might be noted that judgment is required in establishing the canonical graphs, and some vocabulary words may require several canonical graphs, just as they might have several meanings in a standard dictionary.

Once the canonical graphs have been developed, schemata of typical usages needed in the knowledge base can be constructed from the case frames by returning to the sentences they are derived from and joining the canonical graphs representing the case frames. This generalized sentence then serves as a schema for each concept appearing in it. When a large body of such schemata have been collected, they may be summarized or compacted by generalization, as were the case frames to produce the canonical graphs.

Implementation

Although the robotics knowledge base is incomplete, progress has been made in integrating it with the previously written software for the ASPIN system. The dictionary was expanded so that it would include some words pertaining to robotics, along with their grammatical categories. Then, when test sentences were entered into the parser, it was able to successfully generate parse trees. This is the first step towards generating conceptual graphs, and the first step towards using a computer for the design knowledge capture of robotic systems.

Future Work

In the future, the knowledge base will be further developed. With a larger corpus of text, the vocabulary covered by the concept hierarchy and canonical graphs will be expanded, and schemata will be developed. Two items which will merit special attention are the concepts of spatial position (geometry) and continuous time. Neither of these have been needed in the microprocessor domain.

The work thus far has concentrated only on natural language text. Generation of conceptual graphs from diagrams will also be considered, particularly drawings of physical relationships. Since equations are frequent in robotics literature, they must also be considered at some future time.

Conclusion

Progress has been made towards the development of a robotics knowledge base. There is a vocabulary of almost 2000 words. Several robotics concepts and their relationships have been defined. When this knowledge base has been successfully integrated with the tools developed for digital systems, this work should significantly increase the speed and productivity of the entire design cycle, by providing automation for the first stage of design knowledge capture.

References

1. W. R. Cyre, "Textual/Graphical Design Capture for Concept-Level Synthesis," Proc. IFIP on Hardware Description Languages: CHDL'93, Ottawa, Canada, April 26-28, 1993.
2. W. R. Cyre, Integration and Analysis of Requirements with Conceptual Structures, SRC Report C92434, Semiconductor Research Corp., Research Triangle Park, NC, 1992.
3. W. R. Cyre, A Sublanguage Syntax, Control Data Research Report, Control Data Corporation, Minneapolis, MN, July 1988.
4. C. J. Fillmore, "The Case for Case," in e. Bach and R. Harms, eds., Universals in Linguistic Theory, Holt, Reinhart and Winston, New, York, 1968.
5. A. Honcharik, J. Armstrong, and W. R. Cyre, "Mapping Conceptual Graphs to VHDL Descriptions", VHDL International User's Forum, Washington, DC, October 18-21, 1992.
6. IEEE, IEEE Standard VHDL Language Reference Manual, IEEE, NY, 1988.
7. S. Y. Nof, Handbook of Industrial Robotics, John Wiley and Sons, New York, NY, 1985.
8. C. S. Peirce, Collected Papers of Charles Sanders Peirce, A. W. Burks, ed., 8 vols., Harvard University Press, Cambridge, MA, 1960.
9. J. F. Sowa, Conceptual Structures: Information Processing in Mind and Machine, Addison-Wesley, Reading, MA, 1984.
10. Stanford, P. and Mancuso, P., eds. EDIF Electronic Design Interchange Format Version 2.0.0, Electronics Industries Association, 1989.

906225

Cracks, Weld Defects and Porosity Classifiers for Submarine Pressure Hull Steels

#149680

D. Robert Hay[1], J. Helie[1], J.R. Matthews[3], Gaétan Labbé[2], J.F. Boucher[2] and J. L. Kennedy[3]

Abstract

Signal processing and pattern recognition are now playing a key role in meeting the greatly increased nondestructive evaluation (NDE) requirements of advanced materials. These include the use of artificial intelligence in ultrasonics to characterize defects. These methods have been incorporated into APHIUS, the Automated Pressure Hull Intelligent Ultrasonic System, developed for evaluating the integrity of submarine pressure hull steels

A classification study was performed on test specimens taken from pressure hull weldments and plate to demonstrate the ability of APHIUS to distinguish between cracks and volumetric (porosity and inclusions) defects. The data from defects examined in this study provide good separability permitting the design of high performance, user-tolerant defect classifiers for inspection of pressure-hull steel. We can conclude that APHIUS classification can successfully differentiate the target flaw types and is tolerant to different types of cracks (clean and linear vs. "natural" cracks).

Introduction

At present, field inspectors carry out unassisted materials inspections in which each indication is treated as an equivalent type of defect. Accordingly, the materials condemnation limit has been set based on the most serious defect type. The need to identify defects according to type has been a first priority in overhauling outdated submarine materials condemnation criteria. Using flaw identification methods, comprehensive new criteria can be generated which will eliminate wasteful defect removal. As a consequence, submarine refit schedules should be shortened considerably with accompanying cost benefits.

Artificial Intelligence in Ultrasonic Inspection

Use of artificial intelligence in NDE, in particular with ultrasonics to characterize defects, is a rapidly growing technique with novel approaches such as neural networks [1,2], data smoothing and banding [3], and feature-scanning (F-scan) [4] being utilized. Considerable research is also being performed in the field of mathematically modeling the propagation of an ultrasonic wave as it passes through a wide range of structures. In this case, defect identification is to be achieved via mathematical calculation as opposed to statistical techniques.

Efforts have been made, with some success, to distinguish between hidden cracks and no defects in laboratory tests [5]. In another application [4], split spectrum processing was shown to be useful in characterizing the grain morphology (i.e. equiaxial and columnar) and grain size of the material being tested. APHIUS can also be used to study certain frequency dependent properties of a material by employing features in the frequency domain. Pattern recognition techniques have been successfully employed to classify flaws by size and orientation based on time domain features extracted from the ultrasonic signal [6]. A wide range of classification methods (Empirical Bayesian, Linear Discriminant, Nearest Neighbor, and Minimum Distance) and of feature domains (time, frequency, phase, cepstral, autocorrelation, and cross-correlation) were examined in this study to design of classifiers to distinguish among weld defects, cracks, and volumetric (porosity and inclusions) defects in samples of submarine pressure hull weldments. Specimens taken from the pressure hull weldments were combined with a number of fatigue cracked specimens to produce a large sample set for data acquisition.

Artificial Intelligence (AI) is indicated when:

- *there is a lack of skilled people to do the work.*
- *the application has resisted other forms of automation*
- *functions are isolated and performed by an individual or small group.*
- *the working environment may be hazardous to human health*

---

1. Tektrend International Inc., 2755 Pitfield Blvd., St. Laurent, QC, H4S 1T2
2. Directorate of Ship Engineering, Maj. Gen. G. R. Pearkes Bldg., 101 Colonel By Drive, Ottawa, ON, K1A 0K2
3. Defence Research Establishment Atlantic, 9 Grove St., Dartmouth, NS, B2Y 3Z7

Interpretation of ultrasonic signals can vary from operator to operator and human factors such as a training and fatigue influence judgment. An AI system permits faster, more reliable and robust routine interpretation. Also, computer programs can extract more information from data than a human in a reasonable time frame. In NDE, large bodies of knowledge are held by a few experts acquired through extensive inspection experience. Effective extraction and encoding of expert and physical knowledge into a knowledge-based system can disseminate expertise in an application domain or focus it on a particular application. These AI methods can also provide new physical insight into the phenomena involved.

Pattern Recognition and Artificial Intelligence

AI methods simulate certain advanced human functions and interface man and machine with techniques humans use for problem solving in a number of fields:

Pattern Recognition
Intelligent Sensing
Theorem Proving
Knowledge-Based Systems
Natural Language Understanding
Connectionist Systems
Man-Machine Communication
Expert Systems
Computer Vision
Game Theory
Robotics
Decision Theory
Neural Networks
Problem Solving

Of the above, the common *knowledge-based* AI methods for interpretation, diagnosis and analysis fall into two groups, *expert systems* and *pattern recognition methods*. Expert systems formulate a pre-compiled set of rules in a broad plan of attack. Beyond rules, classification provides another paradigm for practical engineering and scientific problems where empirical correlations can be drawn from physical observations. Pattern recognition, matching or classification is the most universally applied generic decision-making task. A human relates a new situation to a larger group of similar cases he has seen in the past. The expert also takes old patterns and creates new ones. The matching process can be simple or involve a classification hierarchy. In many cases, pattern classifier implementation is more practical, rapid and economical than with expert systems.

Pattern recognition is well-suited to interpretation of waveform data generated in ultrasonic inspection and models the methods that humans actually use.

Pattern Recognition

A pattern recognition system, in general, comprises four phases, data acquisition, data transformation, feature extraction, and classification as shown in Figure 1. During development, a training and testing phase is also required.

A *pattern class* is a group of patterns with certain common properties. The objective of pattern recognition is to classify a pattern into a pattern class based on measurements or observations.

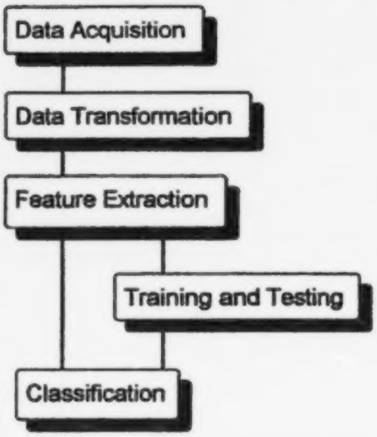


Figure 1. Pattern Recognition System.

Pattern classification methods are well established and largely application independent. Research in classifier design lies in selection of the pattern elements or *features* with powerful pattern representation capabilities. Feature definition and selection is largely application dependent although generic feature sets applicable over a wide range of applications can be defined.

Once the knowledge is organized and formalized using the known identity of the feature classes, it can be used to train the computer to characterize an unknown set of input features or for modifying decision parameters.

Types of Pattern Classifiers

There are several types of pattern classifiers to meet a wide range of applications. The following represent linear, parametric, and non-parametric classifiers:

Linear discriminant function classifier
K-nearest neighbor classifier
Empirical bayesian classifier
Minimum distance classifier
Neural network classifier

These classifiers can be expressed in a convenient canonic form of discriminant functions $g_i(Y)$
where Y is the feature vector,
$i = 1 \ldots c$, and
c is the number of classes

Thus, each class is represented by a discriminant function and a classifier is represented by a set of discriminant functions g_i.

The decision to choose class *i* is made if:

$$g_i(Y) > g_j(Y) \qquad \text{for } j = 1 \ldots c \text{ and } i \neq j$$

All *c* discriminant functions must be evaluated in order to make a decision or a classification.

Linear Discriminant Function Classifier

The Linear Discriminant Function (LDF) Classifier consists of weighted linear combinations of the feature vector components:

$$g_i(Y) = w_{i0} + y_1 w_{i1} + y_2 w_{i2} + \cdots y_n w_{in}$$

where n = number of features in use
y_i = feature vector components

Thus, the LDF classifier defines lines, planes or hyperplanes in the feature space. An unknown sample (described by its feature vector) must fall on the correct side of the i^{th} plane to be classified in Class *i*. This corresponds to the discriminant function with the largest value for the unknown feature vector components.

The LDF classifier is very simple and fast to compute and requires a small amount of computer memory to store the classifier information. On the other hand it can only handle simple feature distributions in the feature space that can be linearly separated.

Minimum Distance Classifier

Each class is represented by the average vector value of all feature vectors in the class. The discriminant functions are the distances from the unknown sample to the average class vectors. The class with the minimum distance from the unknown to its average class vector is the chosen class. This can be summarized as follows:

$$g_i(Y) = \sum_{j=1}^{n} -|y_j - m_{ij}|^2$$

where m_i is the mean feature vector of the ith class

The minus sign makes the discriminant compatible with the general discriminant function statement.
This classifier provides a simple representation of a class in terms of its mean pattern. It is simple and fast to compute but is not robust. Each class will have the same spread regardless of its size or spread.

K- Nearest Neighbor Classifier

The K-Nearest Neighbor Classifier is an extension of the minimum distance classifier. It uses multiple representation points to describe a class instead of just the average vector. To achieve good representation of a class, each class is first split into sub-classes based on natural clustering. Then the average sub-class vectors are computed and stored away for each class. Distances between the unknown and all the sub-class average vectors are computed and only the k smallest distances (or nearest neighbors) and their class identities are kept. Then the unknown is assigned to the class to which the majority of these k nearest neighbors belong.

This classifier provides a more detailed distribution description of each class and is most powerful when handling classes with irregular distributions. It applies internal clustering analysis to each class and thus requires a rather lengthy training session. It also requires more storage to keep track of the sub-class average vectors. Even during the classification phase, it computes all the possible distances between the unknown and all sub-classes, and extensive comparisons to keep the k-nearest neighbors' identity.

Empirical Bayesian Classifier

With the Empirical Bayesian Classifier, each feature is considered independent from the others. The discriminant function is a summation of function of individual features:

$$g_i = \sum_{k=1}^{n} (g_{ik}(y_k)) \qquad i = 1 \ldots\ldots c$$

and $$g_{ik}(y_k) = 1 / [1 + (y_k - m_{ik})^2 / s_{ik}^2]$$

where m_{ik} is the average value of the kth feature for all patterns in class i and s_{ik}^2 is the *standard deviation* of the k^{th} feature for all patterns in class *i*.

This classifier involves the evaluation of simple functions but is not as fast as the previous ones. The classes can have a more complex shape in the feature space.

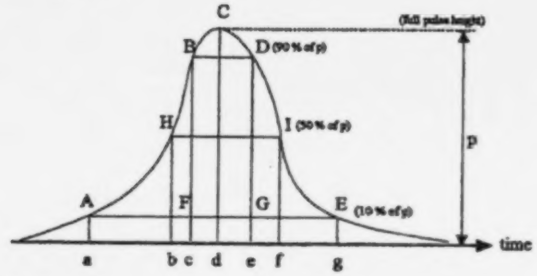
Feature Normalization, Weighting and Selection

Waveform features derived from each signal consist of descriptors of physical measures as shown in Figure 2 ranging from number of peaks, peak location, to energy concentration and correlation index. These feature measurements comprise a large variety of physical units and their corresponding values may differ drastically.

To rationalize these feature elements, each has to be normalized such that they share the same physical unit and their value bounds to a common boundary. This study uses the method of zero mean unit variance mapping to normalize all the features used. The rationalized feature is computed as:

$$x_i = (x_i - m_i) / s_i$$

where x_i is the raw feature value, m_i and s_i are the *mean* and *standard deviation*, respectively, of all x_i in the training set.



Rise Time = AF
Rise Slope = BF / AF
Pulse Location = d
Pulse Width = AE
Peak Width = BD

Fall Time = GE
Fall Slope = DG / GE
Pulse Amplitude = p
Half Max Width = HI

Figure 2. Pulse Shape Descriptors

For a multiple class problem, the global mean will be used in place of m_i while the average standard deviation from all the classes of interest will be used in the equation above. This mapping produces a dimensionless value from the original feature, all features share the same physical dimensions, and the value produced will be loosely bounded to the range (-1,1).

The power of a given feature with respect to classification is really dependent on its statistical distribution. We use a feature analysis mechanism to evaluate each feature using *inter-* and *intra-class distances* and *a Fisher* weighting scheme. Each feature is ranked based on this weighting scheme.

This project used a set of features drawn from the time, power, phase, cepstral and autocorrelation domains [2]. These features have proven effective in a large range of stress-wave-based inspection methods. In any classification problem, use of the full feature set is not necessarily the most powerful in discriminating among signal classes of interest. A simple measure of the ability of a given feature to characterize a class is to study its statistical behavior derived from signals gathered from all the classes of interest in the application. Each *P(xr)* can be estimated from a training set of data using estimators such as:

$$g_r(x_i) = \frac{1}{l_r} \sum_{j=1}^{l_r} f(d(x_i, x_{jr}))$$

A simplified distance function which assumes a unity value if the distance $d(x_i,x_{jr}) < s_i$ and assumes zero value otherwise, where s_i is the standard deviation of the x_{jr}'s within the training set leads to a very simple way of constructing $g_r(x_i)$ in any given boundary *(a,b)*. For any feature value x_i defined within *(a,b)*, the simplified $gr(x_i)$ is equal to the number of class members that are within a distance s_i away. For each pattern class, the corresponding *P(xr)* can be estimated using $g_r(x_i)$ as shown in Figure 3:

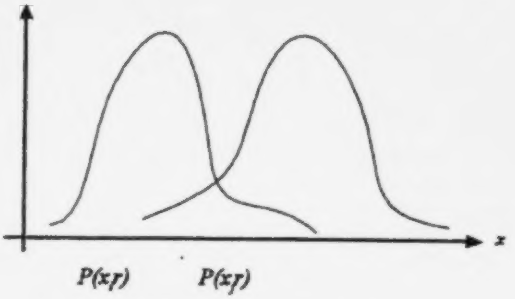


Figure 3. Probability Distributions

The wider apart the distributions, the higher the chances of distinguishing between the classes. For multiple class problems, feature distributions may overlap with one class while being far apart from other classes. Graphical display of each feature distribution contributes as the first-order analysis and evaluation of a given feature.

However, as shown in Figure 4, a combination of two poorly overlapped feature distributions might produce some clearly separated compound distribution mapped on the $ax_i + bx_j$ axis. Similarly, higher-order combinations of feature distributions exist and can be constructed and analyzed accordingly.

Experimental Procedure

The following specimens were employed in this study:

1. A sample from a pressure hull weldment with a cluster of volumetric defects (porosity and inclusions), as shown in Figure 4.
2. 21 fatigue cracked bars with cracks at varying lengths and angles, as indicated in Figure 6.
3. Three plates containing weld defects (Figures 7 through 9).
4. Five bars which contain cracks of varying depths and lengths (Figures 10 through 13).

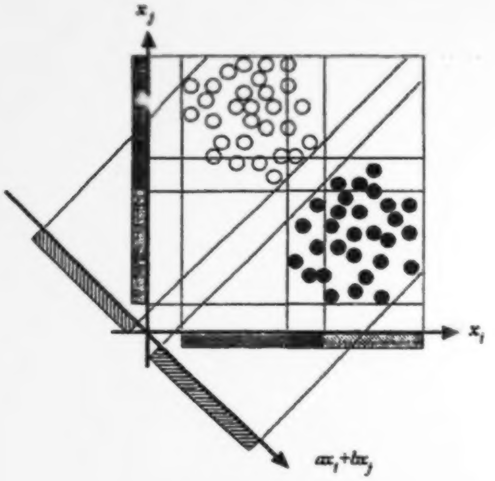

x_j
x_i
ax_i+bx_j

Figure 4. Combinations of Features

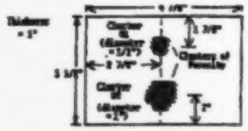

Thickness = 1"
1 3/8"
2 3/4"
1"

Figure 5. Plate containing porosity.

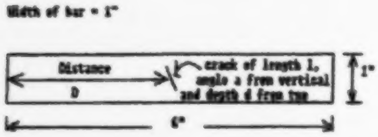

Width of bar = 1"
Distance
D
crack of length L, angle a from vertical and depth d from top
1"
6"

Figure 6. Fatigue-cracked specimens

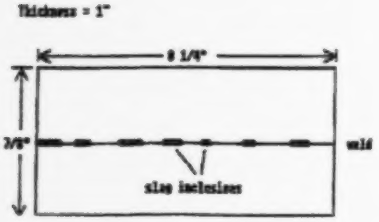

Thickness = 1"
8 1/4"
6 3/8"
weld
slag inclusions

Figure 7. Plate with slag inclusions in a weld.

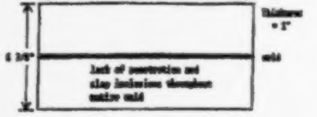

Thickness = 1"
weld
lack of penetration and slag inclusions throughout entire weld

Figure 8. Plate with lack of penetration and inclusions

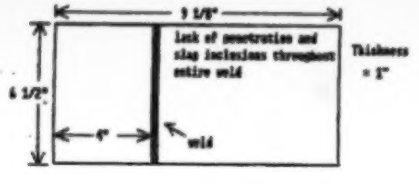

Lack of penetration and slag inclusions throughout entire weld
Thickness = 1"
6 1/2"
4"
weld

Figure 9. Plate with lack of penetration and inclusions

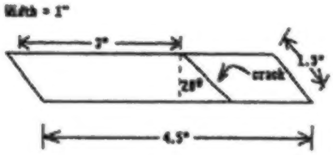

Width = 1"
2"
crack
1.3"
4.5"

Figure 10. Bar with fatigue crack

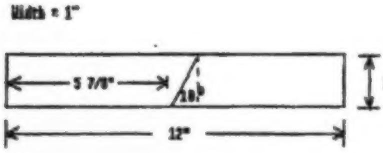

Width = 1"
5 7/8"
1"
12"

Figure 11. Bar with fatigue crack

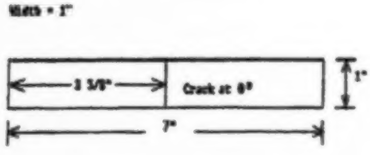

Width = 1"
Crack at 0°
1"
7"

Figure 12. Bar with fatigue crack

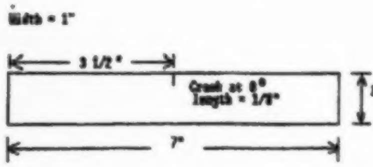

Width = 1"
3 1/2"
Crack at 0° Length = 1/8"
1"
7"

Figure 13. Bar with fatigue crack

Ten signals were collected from each specimen with crack-like defects in the following manner. Five signals were collected from each side of the crack-like flaws. To provide a variation in the type of signal being received for a given flaw representative of that likely to be obtained in in-service submarine inspection, spread in the data for volumetric defects was obtained by taking data at different angles to the defect cluster.

A maximum amplitude signal was obtained and then the transducer was deliberately moved off center and pointed slightly away from the flaw in a random manner to obtain the remaining four signals. In the case of the sample with volumetric defects; since there was only one specimen, 60 signals were taken.

Another example of emulating field conditions is the selection of the sampling window. A very narrow sampling window will filter out non-flaw related noise and allow a "purer" signal to be fed to the classifier. However, in actual inspection a window much larger than the actual signal makes it easier for the operator to get the signal on the screen to analyze or store it. If the window is too narrow, the operator would have trouble finding the signal and centering it in the window. In this study, a movable window which enabled the operator to examine any portion of the signal was employed.

All flaws in this experiment were examined at the single skip distance using a 45 degree Harisonic ABW-0808 wedge. The depth of every specimen studied was 1 inch, so we may compute the skip distance s as two inches from: $s = 2\ d \tan \beta$, where d = depth of specimen and β = skip angle. A single cycle of negative unipolar pulse was applied because its simple sine wave shape lends itself well to signal analysis.

However, before beginning signal collection with the angle beam transducer, a straight beam transducer was used to locate the flaws and the skip distance was marked on each specimen. The straight beam technique proved particularly useful in the case of locating weld defects in the plate. In addition, the straight beam transducer was used to detect any anomalies in the specimen which might interfere with the normal pattern recognition exercise, for instance, excessive porosity at crack tips and double flaws.

To maintain emulation of field conditions, the settings of APHIUS were kept constant throughout the entire experiment. For example, signals from cracks in the bars were received with much higher intensity than those from weld defects in the plates and the gain and/or output voltage could have been adjusted to compensate for this. However, since it is desirable that such adjustments not be made in the field, these settings were left constant at 225 Volts and 49 dB of gain. Another point to consider is that in training classifiers, differences in equipment settings can cause variations in setup to be misinterpreted as actually being associated with the phenomena being studied.

A maximum data length of 2048 points per sample was chosen because there were no particular time constraints for feature extraction in this study. The sampling rate was adjusted to 100 MHz which allows the inspector to move the transducer around easily without losing the signal, and at the same time providing a clear signal free of excessive noise not related to the flaw. A data offset of 1120 points was used corresponding to the time that an ultrasonic signal takes to traverse the sound path to a full skip distance.

CLASSIFIER DESIGN

Three classifiers were designed to examine the feasibility of employing pattern recognition in distinguishing between irregularly shaped flaws. Nine bars and three plates were examined. Five of the bars were actual specimens from a pressure hull and four were fatigue cracked with cracks of varying angles and length. All plates contain weld defects and were taken from pressure hulls.

The first classifier involved a simple two-class problem to evaluate the general effectiveness of APHIUS in analyzing these types of specimens before carrying out further, more detailed studies. For the second classifier, a bar that contains a particularly rough and irregular crack, more typical of field conditions, was used along with a single plate. The third classifier involved three classes of defects (volumetric defects, weld defects, and cracks).

The following procedure was used to prepare data for classification by APHIUS. Features were extracted from all signals and a "feature file" created for each class. Next, the feature files were normalized and split into training and testing sets. The training sets used to design the classifier and the testing sets are used to evaluate its performance. Performance is expressed in terms of a recognition rate which is the percentage of correct classifications. In addition, there was a rejection option which will reject any signal which does not meet classifier requirements for any class. Any feature vector whose probability density function for belonging to a particular class was less than 5% was rejected. If the rejection option is off, all signals will be classified regardless of their similarity to either type of flaw.

Crack and Weld Defect Classifier

Table 2 shows the results of this classifier design exercise. The classifier for this application, which employed the minimum distance classifier, used two features: feature number 74; percent of partial power in the fourth octant of the phase domain, and feature number 113; percent of partial power in the first octant of the autocorrelation domain. The recognition rate of the classifier was 92.45%, with all 15 samples form the weld defects being classified correctly and 4 out of 38 cracks misclassified as weld defects. In this classifier, the rejection option was on, and a signal which was dissimilar to either a crack or weld defect would be rejected as belonging to neither class. Since no signals were rejected, the training and testing sets were relatively well clustered.

Table 2. Results of classification of cracks and weld defects

Classifier Used	Features	Recognition Results
Minimum Distance	74 113	92.5%

Irregular Crack and Weld Defect Classifier

A smaller classification exercise to see how effectively APHIUS could distinguish between weld defects and complex cracking signals was carried out. Most of the signals from cracks in this study showed a very clear sinusoidal shape, which could easily be identified by a human inspector as being associated with a linear crack. For this study a crack which gave a time domain signal which was visually indistinguishable from that from a weld defect was studied. This type of investigation is particularly warranted considering the fact that all cracks in the field will be irregular.

Table 3. Results of classification of irregular cracks and weld defects

Classifier Used	Features	Recognition Results
Linear Discriminant Function	83 92 110 113	100%

As in the preceding exercise, 10 samples were collected from each specimen. Two feature files, one for each class, were created. The crack feature file contained 10 feature vectors corresponding to the 10 signals collected from the bar containing the crack and the weld feature file contained 10 feature vectors corresponding to the 10 signals collected from plate #3. As in the previous classification exercise, the feature files were normalized and split, with the result being two files for each class; one used for training and one for testing.

The final classifier selected was a linear discriminant classifier (Table 3) which employed feature #83, number of peaks above 10% maximum signals amplitude in the cepstral domain; feature #92, percent of partial power in the first octant of the cepstral domain; feature #110, percent of total area under the greatest peak of the autocorrelation domain; and feature #113, percent of partial power in the first octant of the autocorrelation domain. The recognition rate during testing was 100%. The rejection option was employed for this classifier; it can be concluded that the separation between classes is clear.

Crack, Weld Defect and Volumetric Defect Classifier

Data from cracks, weld defects and volumetric defects were organized into three corresponding files. The number of records in these three data files was 87, 29 and 70, respectively. Each record consisted of 2048 points. The results obtained with the classifier designed are shown in Table 4. For this 3-class problem, the classifier consisted of the features numbered 74, 85, 109, 119, and 120. The Minimum distance classifier used resulted in a 98% recognition rate on training and 96% when evaluated on an independent set of data.

Table 4: Results of clasification of cracks, weld defects and volumetric defects

Classifier Used	Features	Recognition Results
Minimum Distance	74 85 109 119 120	Training: 98% Testing 96%

DISCUSSION

Two common features emerge from the classification studies. Feature number 113, percent of partial power in the first octant of the autocorrelation domain, was employed in two separate classification studies. Since autocorrelation measures the periodicity of signals, this shows that the ultrasonic signal reflected from a crack and that reflecting from a weld have distinctly different periodicity characteristics. This may be explained by the fact that a signal from a crack is similar to a sidewall echo, i.e. a clear, easily identifiable sinusoid with an envelope that peaks and decays evenly about both sides. Such a signal will have high periodicity. On the other hand, a signal from a weld defect has no discernible visual periodicity.

Also, the classifiers made extensive use of features in the phase domain (features numbered 61-78). This suggests that different flaw configurations produce different effects on the phase relationships between the frequency components of the ultrasonic signal. A clear signal such as from a straight crack will contain frequency components that are in phase with one another whereas a noisy signal from an irregular defect will contain frequency components whose phase relationships are random.

In these classifiers very few time domain features were employed. This makes the classifier less sensitive to the position of the peak in the data viewing window and provides tolerance to operator flexibility and variability and to setup and coupling parameters.

CONCLUSIONS

The data from defects examined in this study provide good separability permitting the design of high performance, user-tolerant defect classifiers for ultrasonic inspection of submarine pressure hull steels. Classifiers designed in this study are tolerant to:

- *Both artificially (laboratory) generated cracks which are clean and linear and the more irregularly shaped cracks encountered in actual weldments (i.e. natural defect variability).*
- *The position of the signal in the viewing window (i.e., operator variability).*

Weld defects, cracks and volumetric defects can be distinguished from one another under conditions similar to those encountered in the field. In particular, a large variety of cracks of varying depth, length, and angles can be successfully distinguished from weld defects and volumetric defects. Crack-like defects are readily separable from volumetric defects. Even irregularly shaped cracks can be distinguished from weld defects, which indicates that the classifiers designed in this study are good for the wide variety of cracks in the field, in addition to simple linear cracks artificially generated in the laboratory.

REFERENCES

1. "Recent Developments in Quantitative Ultrasonic NDE of Composites", W. Sachse et al., Ultrasonics 1990, Vol. 28, March 1990.

2. "New Neural Networks and Adaptive Pattern Recognition", Internal Report, Tektrend International, 1991

3. "Automated Ultrasonic Defect Discrimination", P. Walkden, Materials Evaluation, September 1989.

4. "Utility of Feature Mapping in Ultrasonic Non-Destructive Evaluation", J.L. Rose et al., Ultrasonics 1988, Vol. 26, May 1988.

5. "Tutorial on Signal Processing and Pattern Recognition", C.H. Chen, Non-Destructive Testing (Proc. 12th World Conference) edited by J. Boogard and G.M. Van Dijk, Elsevier Science Publishers B.V., Amsterdam, 1989.

6. "Time Domain Feature-Based Classification by Size and Orientation of Near Field Flaws", G.S. Kohne and P.J. Coyne, Materials Evaluation 46, August 1988.

905C20

Multiple Views of Knowledge in Diagnosis

Suhayya Abu-Hakima
Knowledge Systems Laboratory
Institute for Information Technology
Building M-50, Montreal Road
National Research Council of Canada
Ottawa, Canada
K1A 0R6
email: suhayya@ai.iit.nrc.ca
tel: (613) 993-8551
fax: (613) 952-7151
NRC #37097

#149682

Abstract

This paper argues that automated knowledge acquisition for diagnosis has had limited success in both failure-driven diagnosis and model-based diagnosis. The paper describes fault-based and model-based reasoning for diagnosis and surveys some of the approaches to knowledge acquisition in both areas. The Diagnostic Remodeller (DR) algorithm I am currently implementing for the automated generation of behavioural models from fault-based knowledge is presented. An example of fault-based knowledge from the Jet Engine Troubleshooting Assistant (JETA) is used to demonstrate how a behavioural model can be extracted with DR.

Fault-Based Diagnosis

Fault-based reasoning (FBR) is used in many diagnostic systems. Knowledge in FBR is largely based on maintenance manuals and interviews with experts intended to capture heuristic knowledge about the maintenance and repair of a device or process. The knowledge in these systems is often represented as hand-coded rules or frames which are organized into troubleshooting hierarchies. At the top level of the hierarchy is the general knowledge representing a problem with the device. This general problem is refined systematically until the leaf nodes of the hierarchy which represent physical repairs to the device are reached. Once these repairs are achieved by a human technician some diagnostic systems re-test to confirm that the symptoms and diagnosed faults are cleared through backtracking in the hierarchy.

FBR systems have evolved considerably since the development of MYCIN [Scott et al. 77]. MYCIN was developed to provide advice treatment for microbial infections. The MYCIN programs started with hand-coded rules which later evolved into meta-rules in NEOMYCIN to provide some structure to an otherwise flat knowledge base [Clancey 86]. The MYCIN approach remains a very widely used approach in FBR systems as described in the literature review of [Abu-Hakima 93].

Diagnosis is often referred to as a classification problem. Chandrasekaran and his colleagues developed MDX, a system that diagnosis a form of liver disease, cholestasis [Chanrasekaran et al. 79]. MDX has a diagnostic hierarchy which is referred to as a conceptual hierarchy since it guides the reasoner globally through diagnoses clustered as concepts that establish local contexts. Local uncertainties and hand-coded knowledge represented in frames are used to guide the diagnosis [Chandrasekaran and Tanner 86]. MDX has served as a model for many well-structured diagnostic systems including RATIONALE [Abu-Hakima 88] and JETA [Halasz et al. 92].

RATIONALE is a workstation diagnosis system that reasons explicitly so that it may support the user with sophisticated explanations of diagnoses that help justify system behaviour and clarify reasoning. In it many of the ideas advocated by Chandrasekaran for structuring FBR systems and handling uncertainty are applied. This approach was found to be ideal for explicitly representing knowledge so that it may be explained [Abu-Hakima and Oppacher 90]. RATIONALE diagnoses faults with Xerox workstations. It generates dynamic and static template-based explanations that include why, how and what-if responses. Explanation remains a major objective of FBR systems and most systems have why and how explanation but do not necessarily generate hypothetical (what-if) ones. RATIONALE's knowledge is in hand-coded frames.

The Jet Engine Troubleshooting Assistant (JETA) is a tool developed to assist a technician in diagnosing aircraft engines using a hypermedia interface which provides contextual help. For a diagnostic application to properly support hypermedia, one requires a structured manner by which to represent the knowledge, reason about it interactively, display it dynamically and explain it to the user (see [Abu-Hakima et al. 93] for a thorough description of JETA's hypermedia interface). JETA's knowledge representation and reasoning strategies are more flexible than those of other diagnostic systems including RATIONALE's.

JETA's troubleshooting knowledge is represented as a diagnostic network that is hierarchical in nature. Each node in the network corresponds to a decision point in the troubleshooting process and the links represent relations directing the flow of control between nodes. The overall network is much broader than it is deep since there are many components and associated symptoms. The number of nodes along a network path varies from four to twelve in a network of approximately 200 nodes. Possible next moves in the network are represented as children of a node. Any node can have multiple parents since a component malfunction may be due to many causes. The troubleshooting knowledge is hand-coded at each diagnostic node as a frame using a custom command language. In JETA as in RATIONALE, advice generating slots are included in the frame and their contents are output to the user as diagnoses or procedures to follow to find a fault. In JETA, advice is supported with a schematic or a graph. An indexed database of schematics and graphs is kept so that only pointers to the database are kept in the frame. The current implementation of JETA links text, graphs and schematics.

Model-Based Diagnosis

Model-based reasoning (MBR) for diagnosis concentrates on reasoning about the expected and correct functioning of a device. A device is modelled based on its components and their expected behaviour [Hamscher and Struss 90]. Such models range from quantitative ones to qualitative ones and all attempt to approximate device behaviour as accurately as possible. Once a device model is stabilized then a device's observed behaviour can be predicted from the model. If a discrepancy in behaviour is detected then possible candidates based on assumed component faults are generated. These candidates are generated based on assumptions that describe correct model behaviour. Sequential diagnosis is used to choose observations, augment a prediction for the candidate faults and update the list of candidates until a dominant candidate is found.

In MBR there are many conflicting definitions for models. They range from causal models represented as semantic networks with links specifying the relations between component nodes to full blown numerical simulations for complex systems and processes that have taken decades to perfect. Generating models is a key problem in MBR. Some researchers generate causal models, others generate models with structure and behaviour while others generate functional models for devices. Knowledge in models has thus far been hand-coded by experts that understand device component behaviour and function.

Davis was one of the earlier proponents of MBR. In [Davis 84] he describes a theory to exploit reasoning on the basis of device structure and behaviour. He defines paths of causal interpretation. He also describes constraint suspension used to identify which components are responsible for which faults. He argues that we need to balance complexity versus model completeness in diagnosis thus we need to enumerate and layer categories of failure. Quite a bit of work has followed Davis' examples and theories.

De Kleer and Williams published a key paper on MBR for diagnosis describing GDE, the General Diagnostic Engine [de Kleer and Williams 87]. GDE infers behaviour from device structure and functionality. It is applied to digital circuits and makes use of an ATMS (Assumption-Based Truth Maintenance System). This work forms the cornerstone of ATMS-based model-based reasoning systems. It was followed by many papers that criticized the approach as not computationally practical in diagnosing faults with large complex systems. Some of the papers criticizing GDE propose the use of hierarchical fault-based reasoning to reduce the computational complexity of de Kleer and Williams' approach. Struss has developed GDE+ which handles: simple dynamic aspects, multiple tests, hierarchical knowledge and unreliable observations [Struss 89]. GDE+ is a migration back to heuristic or empirical diagnoses using fault-based reasoning. Struss points out that neither GDE nor GDE+ address: changing device structures, complex temporal behaviour (feedback), uncertainty or the use of qualitative models in reasoning. In [Struss and Dressler 89] the authors advocate the representation of a fault view for each component. They point out that a fault and a healthy view (state) for a component cannot be true in the same time instant (consistent belief rule). They also give the 'no good inference rule' where the node and its opposite which represents a fault cannot be true at the same instant. The ATMS is then modified to reason with the fault as well as the no-fault behaviour of a device. Their work gives excellent insight into combining model and fault-based diagnosis to deal with GDE's shortcomings.

Other MBR authors have argued about the definition of device functionality versus behaviour. Sticklen in [Sticklen et al. 88] describes modelling a device's functionality by:

- decomposing the device into sub-devices,
- stating abstractly the functions, goals and purpose of the device and
- representing the manner of achieving the device functions, goals and purpose.

A good definition of functionality is one which argues that function is the set of goals the device is intended or designed to achieve [Malin and Liefker 91].

Automatic Diagnostic Knowledge Acquisition

Machine learning is a key approach in knowledge acquisition for diagnosis. Machine learning includes empirical and analytic learning. Empirical learning focuses on learning for classification (including learning rules from real or simulated data for diagnosis). Analytic learning addresses learning for problem solving tasks which include planning, diagnosis, design, natural language understanding, control and execution. There has been an explosion of work in machine learning in recent years. It is viewed as one of the key approaches of reducing the knowledge acquisition bottleneck [Boose 91; Gaines and Shaw 91].

The MOLTKE (MOdels, Learning and Temporal Knowledge in Expert systems) testbed for diagnosis under development at the University of Kaiserslautern, Germany is described in [Althoff et al. 90]. The system is designed to acquire device knowledge for diagnosis. It has an MBR mechanism for acquiring device models based on their components. A *component* of the model includes a name, ports to other components (with optional test costs), possible internal states (with optional test costs), behaviour of the component (either in state tables or rules that represent the constraints the component sets up between its ports and states), subparts and their interconnections (if the component is non-atomic), typical malfunctions with name and effects (model typical behavior when the component fails) and a priori probability of failure). MOLTKE uses case-based reasoning to acquire and refine knowledge that is generalized to a fault-based hierarchy. It also uses explanation-based learning to refine the rules in the fault-based hierarchy to get the minimum reasoning paths for a solution. MOLTKE has been applied to a Computerized Numerical Control (CNC) machining center. It is also under investigation for the problem of driving mining machines.

Carnes and Fischer describe the use of supervised learning for the placement of sensors in a thermal model [Carnes and Fisher 92]. They use cluster analysis using COBWEB to group together observations for diagnosis. An explicit model of diagnosis (a model-based simulation) is used to place the sensors by generating the various states that need to be measured for design or diagnosis. The approach is novel however sensor placement is not a new issue tackled by machine learning [Abu-Hakima 93].

Scotty is an expert system under development at Rocketdyne for the Space Shuttle Main Engine (SSME) [Modesitt 90]. It is aimed at automating the analysis carried out on firing the shuttle engines. The FBR expert system is based on the experience of propulsion engineers and it consists of 125 manually derived rules and 1400 automatically induced rules from example runs which are embedded in a distributed software environment. The rules are poorly structured and will be moved to a relational database management system (RDBMS) to improve the knowledge representation. The author is also planning to enhance Scotty's graphics and generalize the approach to other Rocketdyne engines. Rule induction from data as a technique is not new but this integration with FBR knowledge and its application is somewhat novel.

ACES (Attitude Control Expert System) diagnoses anomalies in the attitude control system of the DSCS-III satellite [Pazzani 90]. ACES is fault-based (rules represented as Prolog predicates). A fault is confirmed or denied by comparing the observed behavior to that predicted with a simulator. In the case where the simulation denies the fault, the heuristic that proposed the fault is expanded to include the tests that the simulator performed to rule out the fault. ACES uses explanation-based learning to identify the conditions under which the heuristic will propose a fault that is denied. The author concludes that failure-driven learning finds sufficient conditions for ruling out a fault and success-driven learning finds sufficient conditions for establishing a fault (but not necessarily ruling others out). Pazzani's work is novel and very relevant to the refinement of fault-based knowledge using model-based reasoning and explanation-based learning.

Pearce describes two parallel approaches to acquiring knowledge for FBR diagnosis [Pearce 88]. The first is generated by a knowledge engineer who hand codes rules form an expert. The second uses examples of failure from a simulation and applies AQ, Quinlan's induction algorithm, to the data. The author concludes that the second approach gives much better results. The only drawback to the use of induction of rules from simulated data is that one needs a good simulation of a device to generate the examples from which the expert system is induced. It is not always possible to find or generate such simulations.

The collection of jet engine sensor data and its interpretation while comparing multivariate linear regression and instance-based learning is described in [Turney and Halasz 92]. The results of the study indicate that instance-based learning can be the basis for a useful diagnostic tool for aircraft engine technicians. The application of instance-based learning for generating libraries useful in jet engine diagnosis is novel. Such an algorithm could be integrated with the Jet Engine Troubleshooting Assistant [Halasz et al. 92] to provide a technician with an on-line sensor monitoring and jet engine diagnostic tool.

There has been tremendous activity in machine learning in recent years. In empirical learning classification algorithms such as ID3 and AQ have been used to induce diagnostic rules from real or simulated data. Classification learning extracts rules from positive and negative examples. In analytic learning explanation-based learning (EBL) has been used in the form of speedup learning to generalize diagnostic rules and shorten reasoning chains. I believe that neither classification nor EBL addresses the problem of *knowledge-rich* learning where structured knowledge is learned. Such rich knowledge would result from learning to produce hypothesis hierarchies such as those described in fault-based reasoning. In addition, learning from structured knowledge to produce new knowledge, such as learning a device model from its fault hierarchy has not been addressed. Learning complex structures especially for diagnosis is by no means an easy problem but it is one that needs to be further addressed by a combination of researchers in both the machine learning and diagnosis fields. Some researchers which have combined learning (empirical or analytic) with FBR and MBR have met with more success as exemplified by some of the complex problems above. I believe that the key to resolving the knowledge acquisition bottleneck in diagnosis lies in the integration of various machine learning algorithms with partially hand-coded knowledge bases used in FBR or MBR.

Automatic Generation of Behaviour from Fault Knowledge: the *DR* algorithm

Hypothesis

Humans use failure-driven reasoning for successful device diagnosis and repair. As humans reason they build primitive mental models of the devices they diagnose and repair. The hypothesis for the *DR* algorithm is that knowledge of failure and repair embodied in most structured diagnostic knowledge-based systems can be used to derive rudimentary device models. The *DR* algorithm will extract rudimentary device models from fault knowledge. The device model represents structure and behaviour and is driven by device functionality.

Motivation

A great deal of effort is expended hand-coding complex knowledge bases for diagnostic FBR. The artifacts these diagnostic systems are developed for are often expensive machines which have been designed and continuously modified so that no existing accurate schematic or design of their behaviour remains. The J85-CAN-15 is a jet engine which is the first application of JETA. The J85-CAN-15 engine was designed in the 1950's and has easily had at least one modification a year since its launch. As a result of modifications and stresses of daily use (flying in the arctic and flying in desert heat) the jet engine is a very different device than was originally designed and sometimes displays inexplicable behaviour. No existing design schematics can completely capture the engines's behaviour. It is also a very difficult device to diagnose. For these reasons a tool such as JETA was developed. As is typical with FBR systems, JETA does not diagnose novel faults. Learning the device component model, its behaviour and functionality using the FBR knowledge provides the technician with a tool that can achieve model-based diagnosis. For these reasons it was concluded that the *DR* algorithm should be implemented.

Background

If we follow the de Kleer [de Kleer and Williams 87] approach which represents a device with functionality as a set of components with behaviour. The device can be diagnosed by assuming a faulty component and enumerating the behavioural states that the fault propagates in the remainder of the device. This is compared to the behaviour that a technician is observing in attempting to isolate a problem. Model-based diagnosis can detect novel faults since the behaviour of the device is the basis of its knowledge representation and reasoning. Fault-based reasoning uses the faults of a device rather than its actual behaviour, hence it cannot detect novel faults. However, model-based reasoning can lead to a combinatorial explosion in producing a diagnosis for complex systems (for example, an aircraft engine) and it does not lend itself to causal explanation.

I am currently implementing the *DR* algorithm intended to address the automatic generation of a functional model of a device from its fault knowledge. That implies the automatic generation of MBR knowledge from FBR knowledge. By extracting a functional model both fault and model-based diagnosis can be pursued in a single system gaining from the advantages of the two approaches while minimizing the disadvantages. The *DR* algorithm is being applied in the area of complex electromechanical devices, specifically jet engines.

Objectives of DR Algorithm

DR is an algorithm that would take as input the fault knowledge of a device. It may also be necessary to take as input some background knowledge related to the device to attempt to learn its full component structure and connectivity. *DR* initially extracts from the fault knowledge base all references to device components and subsystems. Given these components the algorithm backtracks through a diagnostic hierarchy of nodes to generate hypotheses for component connectivity. To further establish component connectivity, *DR* examines symptomatic or parametric knowledge that activates the diagnostic nodes. Symptomatic knowledge is knowledge of device failure which can be used to generate hypotheses about correct device function. This knowledge will be used to derive behavioural knowledge between components.

Approach to the *DR* Algorithm

The top level design of the *DR* algorithm is shown in Figure 1. Two phases clearly divide the design of the algorithm. In the first phase, I propose the use of an existing knowledge base that diagnoses a complex electromechanical system as input to *DR*. The Jet Engine Troubleshooting Assistant (JETA) is a system implemented to diagnose faults with aircraft engines [Halasz et al. 92]. Background knowledge referred to as a zero-based model of the device will be the second input. This background knowledge is general in nature, for example it could include knowledge that a pump delivers some liquid from a sink to a source and needs a pressure to increase or decrease the flow of liquid. It could also include some knowledge about feedback control in moderating the flow of a liquid to a source based on the level of the liquid at the source. It may also include some relational knowledge about Pressures, Temperatures and Volumes as defined in Physics.

To achieve the knowledge-rich learning proposed as the output for *DR* one requires the use of a structured and explicit knowledge representation that can adequately represent diagnostic causality. I propose to use the RATIONALE [Abu-Hakima 88] knowledge representation which is explicit and provides diagnostic causality for the representation of the learned model. The second phase of the *DR* algorithm compares the functional knowledge of a device exemplified in its device model of components and their behaviours with the original fault knowledge. The purpose of the second phase is to look for inconsistencies and gaps in the knowledge. The gaps discovered in the fault knowledge could then be used to diagnose novel faults.

The objective of the *DR* algorithm is to discover and refine a behavioural model of a device whose fault knowledge is represented. In the most general sense the algorithm must identify the components of the device, generate links between those components and generate hypotheses for the behaviours between the components. To achieve this the *DR* algorithm must:

1. identify the end nodes (components) in hierarchy
   -these are nodes that have no child or sibling refinements
   -these are nodes explicitly labelled as replace or repair-type in JETA as they explicitly refer to device components
   -tag appropriate nodes as physical components based on semantic information
2. identify the end nodes in the diagnostic hierarchy related to subsystem to be modelled
   -perform a match with the name or any acronym that could match the subsystem
3. identify the inheritance (direct) links between nodes
   -backtrack from end node to parent node and tag
   -tag shared parents of a node
   -tag siblings of a parent
4. identify the indirect links between nodes
   -tag nodes that activate other nodes without a familial link
   -tag nodes that activate other nodes through the evaluation of parameters
5. hypothesize about the relations (behaviours) between the nodes
   -cluster nodes related by direct or indirect links
   -use semantic information (node names or parameter ranges/values) to relate nodes
6. output the device model for verification to the user
   -map out the identified components of the subsystem
   -relate the components through the hypotheses

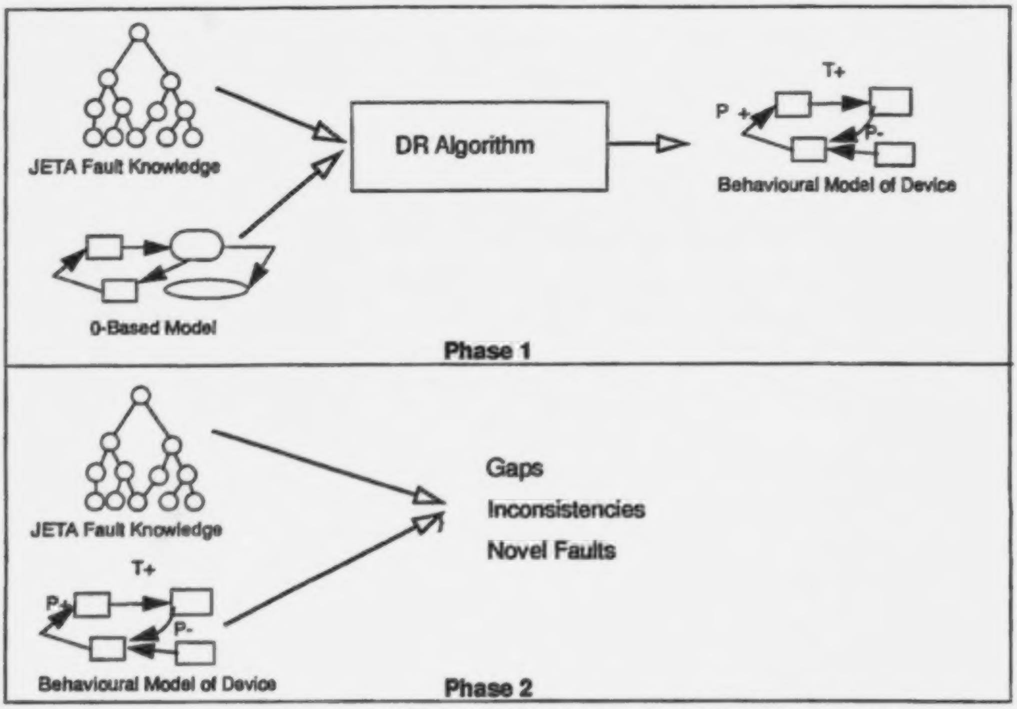


FIGURE 1. Diagnostic Remodeller (DR) Algorithm Design

Extracting a Model from JETA's Fault Knowledge: An Example

An analysis of the JETA fault knowledge shows layers of knowledge which can be visualized as diagnostic trees. The topmost layer is an entry point to jet engine faults and subsequent layers organize the faults into various branches. The phases of operation branches lead to various symptomatic nodes labelled as snags. These snags in turn are refinable down to repair and replacement nodes which form the bottom layer of the diagnostic hierarchy[1]. If one examines the knowledge encoded in these bottom nodes more closely one discovers that they represent faults directly on physical engine components. These physical component fault nodes can be grouped into those affecting one of thirteen subsystems by their nomenclature. One can follow the six steps of the *DR* algorithm introduced above to discover the behavioural network for the main fuel system of the jet engine:

Step 1

One can identify 9 replace nodes through the JETA node frame slot 'node-type'.

Step 2

If one takes a specific subsystem, the MFS (Main Fuel System), one can extract the names of 5 fuel system replacement nodes by pattern matching with the node nomenclature *N-MFS-XXX (this is an internal representation that was used by the knowledge engineer to distinguish between nodes):

1. main fuel control (MFC)
2. overspeed governor for MFC (OSG)
3. MFC/ABFC signal line
4. main fuel pump supplying MFC
5. a pressurizing and drain (P+D) fuel valve

Step 3

For each of the 5 replacement nodes direct single and shared inheritance links can be traced, for example:

- the MFC and MFC/ABFC signal line nodes share the parent node no hesitation at acceleration area

---

[1] The diagnostic hierarchy is referred to as a network since it includes relations that are not directly inherited that allow the JETA reasoner to jump around between nodes thus forming more of a network than a hierarchy.

- the P+D valve and main fuel pump share the engine flameout parent node
- the main fuel control (MFC) delivers fuel to the engine fuel nozzles

Step 4

Indirect links between the subsystem nodes can be identified by activation rules and evaluation of parameters.

Step 5

A causal topological network can be the basis for hypothesized component-behaviour relations. Nodes are clustered based on direct or indirect links. They can also be clustered based on semantic knowledge such as names and ranges of parameter values.

Step 6

The hypothesized device model that includes component enumeration and behaviour can be output to the user for verification. It can also be tested by 'breaking' the correct modes of operation as described for Phase 2 of the *DR* algorithm.

As a device model of hypotheses that link JETA's components is formed it is important to explain *DR*'s results. This can be achieved by mapping the newly learned components and behaviours into a set of RATIONALE hypotheses so that the explanation strategies can be output. This can be achieved by mapping JETA's knowledge representation for the causal component nodes to RATIONALE's.

Issues

There are five main issues that I will need to answer in the implementation of the *DR* algorithm. The first is what is the exact form of the learned model when some or no background knowledge is used. If no background knowledge is used is the model much more than a causal model rather than a component-behaviour model? The latter model is required to diagnose novel faults. Also, what is the minimum background knowledge required to learn a device model from FBR knowledge. That can be extended to explore whether background knowledge alone provides one with a device model. Can this model in turn be used to derive FBR diagnoses?

A traditional simulation of a device or process relates to a device model. A key question is what is needed in addition to the simulation to generate a device model or a FBR model which in turn can be used for diagnosis?

The *DR* algorithm will be implemented as a general algorithm useful in generating models for devices other than jet engines. It is not obvious that its background knowledge will make it specific for generating a particular model. However the FBR knowledge used as input will make it specific to generating a model for a particular device. One question could be whether or not the algorithm can be generalized further so that it diagnosis abstract (e.g. software) versus physical (e.g. jet engine) systems.

Finally, the *DR* algorithm requires a highly structured FBR knowledge base. One key question is what criteria will allow it to extract a causal model from a rule- versus a frame-based fault model.

Conclusion

This paper argues that automated knowledge acquisition for diagnosis has had limited success in both failure-driven diagnosis and model-based diagnosis. The *DR* algorithm for the automated generation of causal models from fault-based knowledge is introduced. An example of fault-based knowledge from the Jet Engine Troubleshooting Assistant (JETA) is used to demonstrate how a causal model of the main fuel system of a jet engine can be extracted with *DR* from the fault knowledge.

References

Abu-Hakima, S. [1988],
RATIONALE: A Tool for Developing Knowledge-Based Systems that Explain by Reasoning Explicitly. Masters thesis, Carleton University, Ottawa, Canada. May 1988.

Abu-Hakima, S., and Oppacher, F. [1990],
Improving explanations in knowledge-based systems: RATIONALE. Knowledge Acquisition journal (2). 301-343. December 1990.

Abu-Hakima, S. [1993],
Diagnostic techniques in knowledge-based systems: a review of approaches, applications and issues. 100 p. To be published as an NRC ERA (1993).

Abu-Hakima, S., Halasz, M., and Phan, S. [1993],
An Approach to Hypermedia in Diagnostic Systems. To appear as Chapter 10 in Intelligent Multimedia Interfaces, Edited by Mark Maybury, AAAI Press (1993).

Althoff, K-D., Maurer, F., and Rehbold, R. [1990],
"Multiple knowledge acquisition strategies in MOLTKE." *Current trends in knowledge acquisition*, Published by IOS, Amsterdam, Netherlands. pp. 21-40, (1990).

Boose, J.H. [1991],
"Knowledge acquisition tools, methods and mediating representations", Knowledge Acquisition for Knowledge-Based Systems, Edited by H. Motoda, R. Mizoguchi, J. Boose and B. Gaines. Published by IOS Press, the Netherlands. pp. 25-62, (1991).

Carnes, J.R. and Fisher, D. [1992],
"Inductive learning approaches to fault isolation and sensor placement", *Notes of the Third International Workshop on the Principles of Diagnosis*, October 12-14, Rosario, Washington pp. 132-138, (1992).

Chandrasekaran, B., Gomez, F., Mittal, S. and Smith, J.W. [1979],
"An approach to medical diagnosis based on conceptual structures". Proceedings of the Joint Conference on Artificial Intelligence, 134-142.(1979).

Chandrasekaran, B. and Tanner, M.C. [1986],
"Uncertainty handling in expert systems: uniform vs. task-specific formalisms". In Kanal L.N. and Lemmer J., editors, Uncertainty in Artificial Intelligence, North Holland Publishing.(1986).

Clancey, W.J. [1986],
"From GUIDON to NEOMYCIN and HERACLES in twenty short lessons: ORN final report 1979-1985". The AI Magazine, pp. 40-60,(1986).

Davis, R. [1984],
"Diagnostic reasoning based on structure and behaviour", Artificial Intelligence, Vol. 24, pp. 347-410 (1984).

de Kleer, J., Williams, B.C. [1987],
"Diagnosing Multiple Faults", *Artificial Intelligence*, vol. 32, (1987).

Gaines, B. and Shaw, M. [1991],
"Foundations of Knowledge Acquisition", *Knowledge Acquisition for Knowledge-Based Systems*, Edited by H. Motoda, R. Mizoguchi, J. Boose and B. Gaines. Published by IOS Press, the Netherlands. pp. 3-23, (1991).

Halasz, M., Davidson, P., Abu-Hakima, S., and Phan, S. [1992],
JETA: A Knowledge-based Approach to Aircraft Gas Turbine Engine Maintenance, Journal of Applied Intelligence, 2, pp. 25-46, Kluwer Academic Publishers, NRC 31832, (1992).

Hamscher, W. and Struss, P. [1990],
"Model-Based Diagnosis", *AAAI-90 Tutorial Notes*, Received at the eighth national conference of artificial intelligence, Boston, Massachusetts. July 29, 1990. pp. 1-179, (1990).

Malin, J.T., and Liefker, D.B. [1991],
"Representing functions/procedures and processes/structures for analysis of effects of failures on functions and operations", Proceedings of 1991 NASA Goddard Conference on Space Applications of Artificial Intelligence, Greenbelt, Maryland, pp. 141-151, (May 13-15 1991).

Modesitt, K.L.[1990],
"Inductive knowledge acquisition experience with commercial tools for space shuttle main engine testing", *Proceedings of the fifth conference on Artificial Intelligence for space applications*, NASA Marshall Flight Center, pp. 487-496, (May 1990).

Pazzani, M.J. [1990],
"Learning in order to avoid search in logic programming", *Computers & Mathematics with Applications*, 20(9-10), pp. 101-10, (1990).

Pearce, D.A. [1988],
"The Induction of Fault Diagnosis Systems from Qualitative Models", *AAAI-88, Seventh National Conference on Artificial Intelligence*, Saint Paul, MN, (21-26 August 1988).

Scott, A.C., Clancey, W.J., Davis, R. and Shortliffe, E.H. [1977],
"Explanation capabilities of production-based consultation systems". Stanford University, Dept. of Computer Science, Report No. STAN-CS-77-593,(1977).

Sticklen, J., Chandrasekaran, B., and Bond, W.E. [1988],
"Distributed causal reasoning for knowledge acquisition: a functional approach to device understanding", 3rd AAAI sponsored Knowledge Acquisition Workshop for Knowledge-Based Systems, Banff, Canada, pp. 34-1 to 34-18, (1988).

Struss, P. [1989],
"New Techniques in Model-based Diagnosis", *Proceedings of Knowledge-based Computer Systems*, Bombay, India, (11-13 December 1989).

Struss, P., Dressler, O. [1989],
"Physical Negation - Integrating Fault Models into the General Diagnostic Engine", *Proceedings of the 11th International Joint Conference on Artificial Intelligence (IJCAI-89)*, Detroit, MI, (20-25 August 1989).

Turney, P. and Halasz, M. [1992],
"Instance-based learning and multivariate linear regression applied to J85 transient performance data", *Internal Report, Knowledge-Systems Laboratory, IIT, NRC*, pp. 1-25. (1992).

904015

OmniEx - An Expert System for Automated Fault Prediction and Diagnosis

Jocelyn Boily, P.Eng.; Barry Graham[1], P.Eng.; Bernard Marrocco, M.Sc.
Spectrum Engineering Corporation Ltd.
P.O. Box 687, Peterborough, Ontario, K9J 6Z8

#149683

Abstract

OmniEx uses expert system, data management and analysis, and operator interface tools to manipulate and maintain sensor data and a knowledge base for a robotic, electromechanical or process engineering system. Capabilities include data monitoring, compression, storage and range analysis; fault prediction, detection and diagnosis; the ability to learn, so that the experience gained from past diagnoses can be used to improve the speed and accuracy of future diagnoses; maintenance supervision and planning; reliability analysis; and reporting through a pointer-based graphical user interface.

An inference engine capable of multiple fault diagnosis is at the heart of the system. A rule based algorithm strategically identifies modules and parts of the monitored system which could be related to a fault, taking into consideration sensor data, stored knowledge, previous faults and, where appropriate, operator commands to infer the actual health of the system and report a list of possibly faulty components.

Fault prediction is afforded by the temporal extrapolation of historical sensor data using regression techniques, with the operator alerted if the projected readings for a particular sensor lie outside a predetermined range.

An application developer automates the population of the knowledge and data bases for a given system to be monitored.

The modular architecture of OmniEx, and the generic nature of the modules, together with the application developer, provide a powerful applications package which can be used to implement an automated data management and fault services system on a broad range of practical engineering systems.

The development work was funded through the Canadian Space Agency's STEAR program.

Historical Background

Canada's contribution to the international space station *Freedom* will be the Mobile Servicing System (MSS). The role of the MSS on the space station will be support of assembly, maintenance, servicing and transportation functions; the retrieval, berthing and deployment of visiting spacecraft; the servicing of attached payloads; and the extra-vehicular activities of astronauts. The mobile robotic manipulator devices of the MSS will be operated remotely, with automation of certain functions.

To meet the long term technology needs of the MSS Program and to maximize the benefits of the program to Canadian industry, a Strategic Technology for Automation and Robotics (STEAR) Program has been sponsored by the Canadian Space Agency. OmniEx provides automated data management services, one of several developing technologies funded under that program.

OmniEx Capabilities

OmniEx is capable of acquiring, compressing and storing real-time sensor and control data, and managing and analyzing those data to perform failure diagnosis, fault prediction, fault history reporting and reliability data gathering.

OmniEx will:

1. interface with the data bus of the monitored control system (MCS), which will provide sensor data, control data and fault conditions;
2. acquire, range validate, compress and store these operational data;
3. when triggered by a fault condition from the MCS, by the operator, or through its own

---

[1] Address correspondence to Mr. Graham.

data analysis, perform fault diagnosis through an inference process using both operational data and on-line system knowledge;

4. display an ordered list of the probable failure mechanisms to the operator via the operator's console;
5. on request, justify the diagnosis by providing the specific rules applied to arrive at the diagnosis; this will assist the operator in confirming the cause of the failure;
6. record the confirmed fault diagnosis in the failure history database;
7. periodically review the sensor data in the operations database and analyze for trends which will give an indication of part degradation which could cause a failure at some future time;
8. use experience gained from past diagnoses to improve the efficiency and accuracy of future diagnoses (that is, provide learning capabilities).

Benefits of Employing OmniEx

The MSS application of OmniEx will increase on-orbit crew efficiency through fault detection and diagnosis. When a fault has been identified, relevant information from the operational database will be analyzed using a knowledge base (KB) and inference engine to diagnose the cause and report a list of the probably failed parts to the operator. This will result in reduced maintenance costs and reduced outage time.

Another application for OmniEx which will reduce operational down time will be health monitoring and failure prediction. Trend analysis of sensor data will provide an indication of the health of parts associated with that sensor. For example, a motor current that is gradually increasing over a period of time could indicate deterioration of the motor bearings or other incipient problem. Maintenance activity could be planned based on these precursor indicators to avoid the cost and inconvenience of an unscheduled outage.

OmniEx will reduce ground support staff requirements by handling some of the routine data processing and analysis tasks. The OmniEx operations database (DB) will contain data acquired from the MSS sensors, as well as control commands issued by the MSS control computers to the equipment. It would be a relatively simple task for OmniEx to analyze the information in the operations DB for individual part usage, and track usage against expected lifetimes. Thus, regular maintenance intervals could be predicted and updated as required, based on the actual hours of operation as well as the in-service time.

While OmniEx was developed to meet future requirements of the space program, the technology and implementation are readily adaptable to a variety of terrestrial applications. Any electromechanical or process system with a high degree of instrumentation and automated controls would be a candidate application for OmniEx. The same benefits described for the MSS would be realized by any application.

System Description

Architecture

Fig. 1 shows the general architecture of OmniEx, a modular program that uses database technology to manage the data and knowledge base technology for fault diagnosis and prediction.

Each block shown on Fig. 1 represents a program module, information base or data stream that can be defined in terms of its functional capabilities, contents and interface protocol.

Overall control, including scheduling of the other modules and message passing between modules, is provided by the Control Module (CM). The Data Analysis Module (DAM) handles incoming sensor and control data, while the User Interface (UI) module provides mouse based graphical interaction with an operator. The other two major components are the Data Management Module (DMM), which performs detailed database management, and the Knowledge Engineering Module (KEM) which provides the fault diagnosis, fault prediction and learning capabilities. The DB is implemented with the commercial Oracle relational database management system (RDBMS).

The KEM has access to a KB in which design and operational information concerning the MSS is stored, along with the knowledge of how parts interact (entity flow) and how they can fail (failure mechanisms). The presence of any particular failure mechanism is assessed through sensor functions which describe the set of sensor evidence which best indicates the particular failure. Fig. 2 shows a partial sample of the KB structure for a single system component.

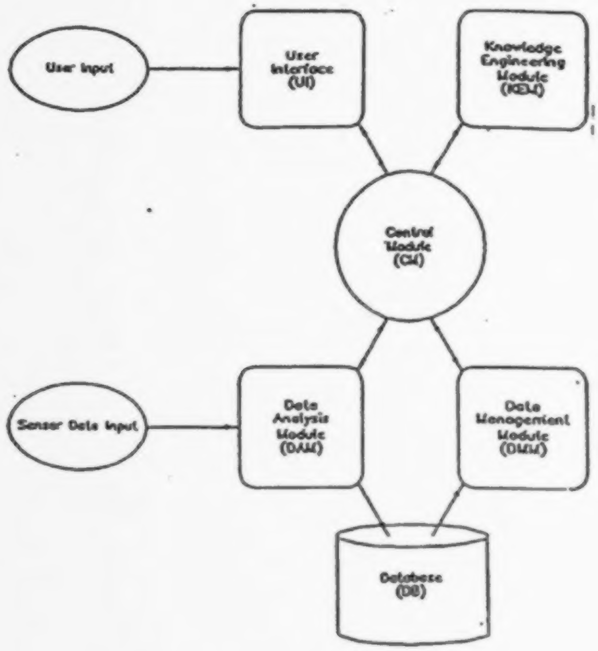

User Input
User Interface (UI)
Knowledge Engineering Module (KEM)
Central Module (CM)
Sensor Data Input
Data Analysis Module (DAM)
Data Management Module (DMM)
Database (DB)

FIGURE 1: Architecture of OmniEx

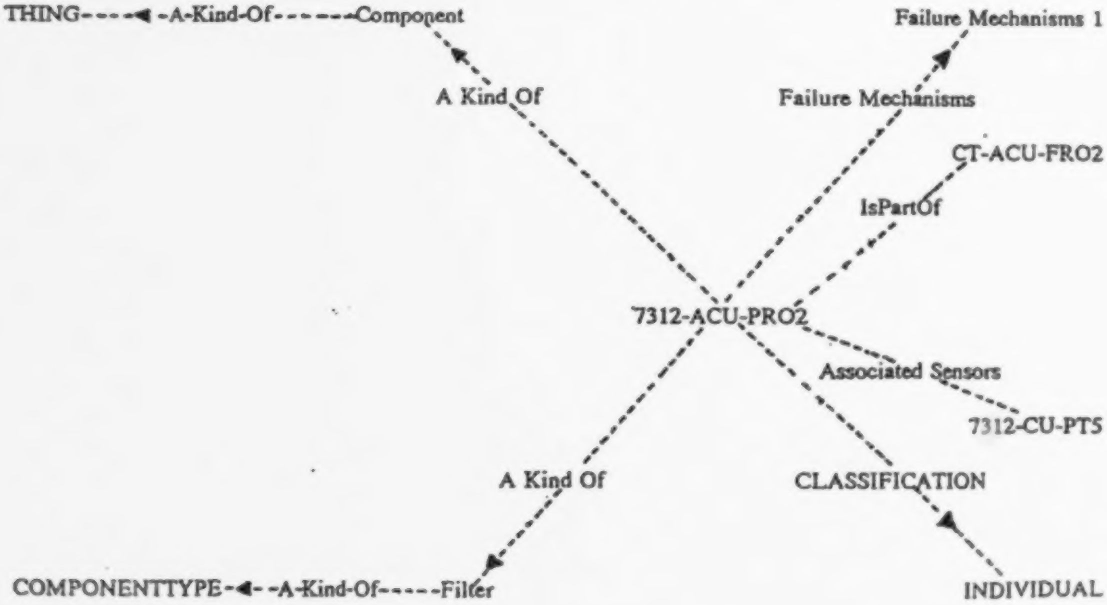

THING
A-Kind-Of
Component
Failure Mechanisms 1
A Kind Of
Failure Mechanisms
CT-ACU-FRO2
IsPartOf
7312-ACU-PRO2
Associated Sensors
7312-CU-PT5
A Kind Of
CLASSIFICATION
COMPONENTTYPE
A-Kind-Of
Filter
INDIVIDUAL

FIGURE 2: Part of the Knowledge Base Structure For An Air Filter (Part Code 7312-ACU-FR02)

OmniEx is a modular system with a clear distinction between modules which are specific to the application, and those which would remain largely or totally unchanged for implementation on any engineering system. Referring to Fig. 1, the CM, DAM, DMM, KEM and UI constitute the core of the generic OmniEx, while the DB and the KB will be tailored to the requirements of the specific application.

An Application Developer (AD) is associated with the generic portion of OmniEx, which facilitates the configuration of OmniEx (specifically, the DB and KB) for a specific project. The AD provides a friendly means of performing a crucial task: the acquisition of computer system knowledge from the human expert(s), and the conversion thereof into efficient computer code. The generic OmniEx system and the AD constitute a powerful applications package which can be used to implement an automated data management and fault services system on a broad range of practical engineering systems.

The CM, DAM, DMM and UI are coded in the "C" programming language for speed, while data storage and retrieval are handled by the Oracle RDBMS (commercially available software). The UI is implemented using the X window system. The KEM is coded in Nial (Nested Interactive Array Language, developed by the Department of Computer and Information Science at Queen's University).

OmniEx is currently implemented on a Sun Microsystems' SPARC station IPX workstation with 16 MB of RAM.

Fault Diagnosis

On fault detection the KEM reviews historical sensor data to provide a list of possible diagnoses, along with rankings[1] and, on request, explanations of its reasoning. The algorithm diagnoses faults by selectively adding to a list of possibly faulty parts, checking each part in this list and adding or removing parts as necessary based on the result of the check. The failure mechanisms (FM) of each part are predetermined by the subject system human expert. Rules, also defined by the human expert, are based on one or more sensor functions and determine whether or not a particular FM has occurred. Diagnosis can be done not only for single faults, but also in those cases where other (known) faults may already exist (i.e. multiple fault diagnosis).

The learning component of the KEM uses the past experiences of the diagnosis algorithm to improve future diagnostic sessions, using case-based reasoning.

Fault Prediction

Fault prediction is afforded by the temporal extrapolation of historical sensor data using regression techniques. If the projected readings for a particular sensor lie outside of a predetermined range, the operator is alerted that a component failure is likely. OmniEx provides details of its findings, including a graphical representation of the actual and expected trends.

The User Interface

Interaction between OmniEx and the operator is handled by the UI module. Fig. 3 illustrates the appearance of the workstation screen during a diagnosis session. The left window, employing both a textual list and a bar graph, lists the possible diagnoses and a numerical estimate of confidence for each listed part. The right window shows a textual history of the "thought processes" of the diagnosis algorithm.

At the top of the left window in Fig. 3 are several "buttons" which the operator may "push" to reveal additional details, such as WHY, which provides and explanation of why a selected part was included in the list of possible faulty parts.

The UI also has windows for the DAM (allowing real time graphical viewing of a sensor's reading), the DMM (which gives the user access to the DB), the KEM (providing graphical browsing of the KB) and an on-line context sensitive help facility. A window will "pop up" whenever the system has predicted the future failure of a part.

Proof of Principle

For development, testing and demonstration, two diverse trial terrestrial applications were selected.

The Fairing Servicing System (FSS) is a robotic manipulator system developed to replace damaged fairings on a variable depth sonar cable on board a Canadian destroyer. The FSS has several manipulators which function together to replace the

---

[1] Ranking is based on confidence estimators, whose functional dependencies include reliability parameters, and the fraction of conditions indicating that a part's failure mechanism has occurred.

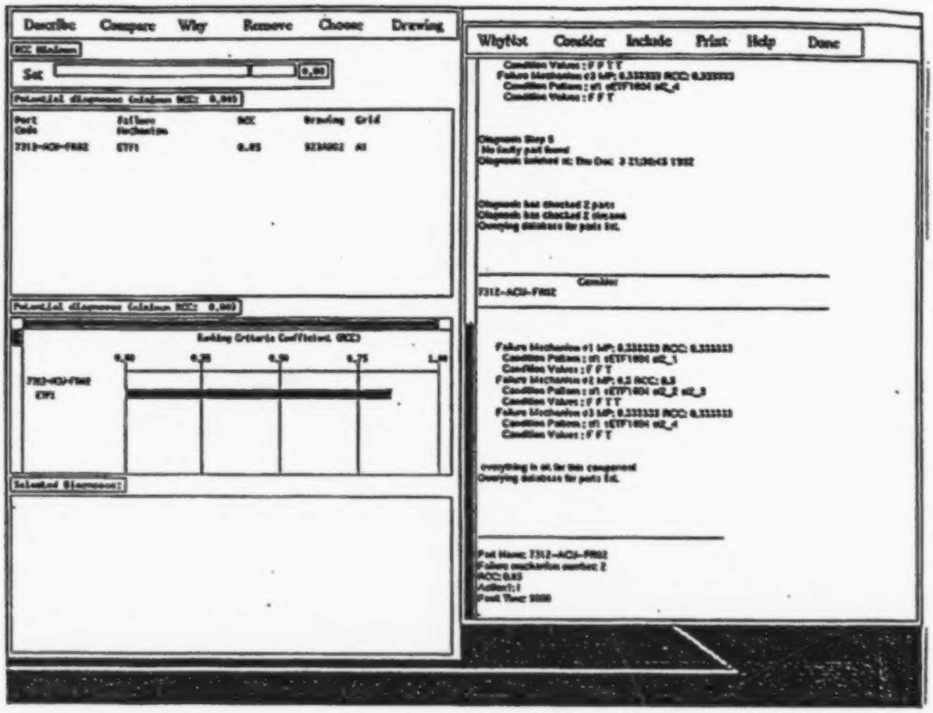


FIGURE 3: OmniEx Screen During a Diagnosis Session

nose and the tail pieces of a cable fairing. These two parts are held together with fasteners which are removed and later replaced. The FSS has a moderate level of sensor density and comprises a fairly complex arrangement of interacting modules.

The Reactor Building Ventilation System (RBVS) for a power nuclear reactor containment building provides filtered, temperature-controlled air to the reactor building, returning the effluent outside after filtering (possibly radioactive) particles. The system include valves to control air flow and to isolate the reactor building if certain abnormal conditions (for example, elevated radioactivity levels) occur, and sensors to monitor the state of the system.

Complete knowledge and data bases were developed for both the FSS and the RBVS. Computer-based simulators of the two systems were used to simulate the output from the sensors and control systems. Output data were used to test the performance of OmniEx modules, including the operation of the UI and the capability of the KEM to successfully identify faults. OmniEx successfully monitored their (simulated) sensor data in real time, providing accurate fault diagnosis and prediction.

OmniEx operator training has also been facilitated by the provision of simulated sensor data.

Future Development

There are several planned upgrades for future versions of OmniEx:

Fault Detection

The Fault Detector will examine sampled and compressed data from the DAM, using a system specific fault detection pattern file to determine when the historical data indicates that a fault has occurred. The Fault Detector, which will be distinct from the existing range analysis function, will then trigger a fault diagnosis.

Maintenance Services

OmniEx will be able to track part usage, providing maintenance planning, scheduling and reporting. Confirmed diagnoses will be stored in the operations planning table of the DB for implementation of repair procedures. Operations data will be analyzed to track usage of parts and update the maintenance intervals in the maintenance table of the DB.

Conclusions

1. The types of engineering knowledge necessary for fault diagnosis have been identified.
2. Relevant thought processes employed by an expert during fault diagnosis have been developed and documented.
3. The relevant system and part engineering knowledge for model mechanisms have been represented successfully using a commercial RDBMS.
4. The system functional information, including power, control and parts flow, was modeled using logic rules and logic functions.
5. The diagnostic process was demonstrated using inference techniques.
6. The modular approach taken for the implementation of the initial concept lends itself to application to a diverse range of subject systems.
7. The planned capabilities of OmniEx have been realized, and their functionality and viability verified through testing with simulated data of real systems.

Acknowledgements

This development work was funded through a contract from the Canadian Space Agency (STEAR Program).

OmniEx was developed in collaboration with the Department of Computing and Information Science at Queen's University, Kingston, Ontario.

406567

Expert System for Digital Circuit Troubleshooting in an Interactive Learning Environment.

K.Sandrasegaran & A.Malowany
Department of Electrical Engineering
McGill University
3480 University Street
Montréal, Québec, Canada, H3A 2A7

#149684

Abstract

In this paper, we present an application of an expert system that can solve troubleshooting problems in the domain of digital circuits. The expert system is to be incorporated within an Interactive Learning Environment(ILE) for the domain of Digital Electronics in which a student or novice is presented with a fault diagnosis problem. The role of the expert system is to solve the troubleshooting problem, if requested to do so by the user, and provide explanations of its actions. The expert system uses a wide variety of information: domain knowledge, application knowledge, observations of the faulty circuit and knowledge that the student is familiar with from classroom instruction and previous troubleshooting exercises. A digital counter application is used to demonstrate the troubleshooting techniques and the knowledge representation issues. The work presented here differs from most expert system and artificial intelligence applications for automated fault diagnosis that solve the diagnostic problem in the most efficient manner.

Introduction

An interactive learning environment is a computer-based simulation in which a novice can be coached or tutored in a particular domain and task. Over the last decade a number of computer-based training programs have been developed in a wide range of domains and tasks. In our work we are building an interactive learning environment for the task of troubleshooting digital circuits.

A digital circuit is an interconnection of electronic circuit components each of which performs a specific function. The function of a component can be observed by measuring the behavior of electric signals at the input and output of the component. The combined functions of the various circuit components in a circuit is used to achieve the designed purpose of the circuit. A digital circuit is faulty if its behavior is different from the normal or designed behavior of the circuit. Troubleshooting is the process of finding the change in the structure (component connectivity and component function) of the circuit which is causing the misbehavior.

A number of interactive learning environments for diagnostic tasks have been built in various domains.

Domain	*Implementation*	*Ref*
Electronics Systems	Heatkit Power Supply	[3]
	F-15 Aircraft Modules	[13]
	Simple D.C Circuits	[20]
Marine Systems	Powerplant of a ship	[9]
	Turbina Vyasa	[19]
	Steam Power Plant	[12]
Satellite System	Ground Station Control	[14]
	Supervisory Control	[5]
Thermal System	Gas heating system	[10] [11]
Medical Diagnosis	Guidon	[4]
	Neomycin	

Why use computer-based environments ?

The basis for use of computers as an education tool is well supported by evidence from researchers in the educational field. Bloom [1] reports that "... conventional teaching, teacher in front of 20-200 students, provides one of the least effective methods for educational delivery.." and " .. one-to-one tutoring is far superior to conventional teaching and the most effective educational delivery method..". Studies on human retention [8] conclude that ".. instruction that utilizes several of the student's senses results in better retention: people retain 25% of what they hear, 45% of what they hear and see and 70% of what they see, hear and do ". Information is better retained if a student is an active participant rather than a passive absorber and this dictates balance of control between student and tutor [17].

Benefits of Interactive Learning Environments

Computer-based learning environments offer a number of advantages especially for the task of troubleshooting. Some of the advantages are itemized below:

1) Computer-based learning environments provide an opportunity for students to practice or do tasks relevant to information gathered in the classroom and hence compile this knowledge.
2) Student attention is heightened by the system interactiveness.
3) Availability of the learning environment at all times of the day and wherever the hardware/software platform is available.
4) Constant human supervision is not necessary because of the resident expert system that provides expert-like advice.
5) Pace of instruction is controlled by the student.
6) An endless variety of problems and exercises can be generated depending on the progress of the student.
7) Feedback provided could be at the level of student knowledge.
8) Student can explore their actions in the learning environment without the fear of real-life consequences such as blowing equipment, component etc.
9) Computer-based learning environments promote discovery learning which is better retained.
10) Fault scenarios which rarely occur in real-life troubleshooting could be easily simulated thus enabling student to handle these situations in real-life.
11) Computers have the capability to offer explanation by analogies and dynamic graphical representations and their graphical capabilities could be used to illustrate concepts that are difficult to visualize on a blackboard or oscilloscope, eg. race conditions.
12) Input from the user during a tutorial session could be used to develop a model of the student's misconceptions and missing conceptions. This could be used for planning tutorial activity and providing explanations which the student can understand.
13) Students problems are cleared as and when they come up instead of waiting for help from a colleague or teacher.
14) Training is possible in remote areas where expertise is not available.
15) The learning environment could evolve into a job-aiding system for real-time on-line assistance.

Why use troubleshooting?

Digital circuit knowledge that is taught to a student in the classroom is compiled when the student applies the knowledge during problem solving. This could be in the form of problems in circuit analysis, circuit design or circuit troubleshooting. We have chosen the latter as a means of assisting students to compile the classroom knowledge. Electronic troubleshooting is an activity that is performed on a real circuit and students' interest is heightened as they are applying knowledge to a real-life application. Operational features of electronic devices such as fan out, thresholds etc. are often left out during classroom instruction. Thus a simulated environment in which these concepts can be introduced gradually forms a good interface between classroom knowledge and reality. Troubleshooting also promotes discovery learning in which students can discover the concepts and laws of the domain by self initiated actions.

Functionality of an ILE

An interactive learning environment(ILE) for troubleshooting tasks has to provide the following functionality:

- simulate the behavior of the domain in which diagnosis takes place under normal and faulty conditions (*Simulator*).
- access to tools and facilities that the diagnostician uses to observe the behavior of the physical system (*Interface*).
- model user performance during diagnosis so as to provide instruction based on the students understanding of the domain (*Student Model*),
- interpret student actions and ascertain the correctness of student actions in the context of the domain and the problem being solved (*Task Model*),
- solve the diagnosis problem for the student (*Expert Model*), and
- generate diagnostic problems in the domain such that each requires application of some new knowledge (*Problem Generation*).

In this paper we shall be concerned with the application of an expert system to solve troubleshooting problems presented to the student in the domain of digital circuits.

Simple Digital Counter Application

A simulation of a simple digital counter system is used in the interactive learning environment as the digital circuit on which troubleshooting actions are performed by the student.

The student can examine the functioning of the digital counter application using the mouse on the interface shown in figure 1. Mouse clicks on buttons cause it to be pressed and mouse drags on switches cause a switch to toggle. The leds (light emitting diodes) and the seven segment element display the outputs of the digital system.

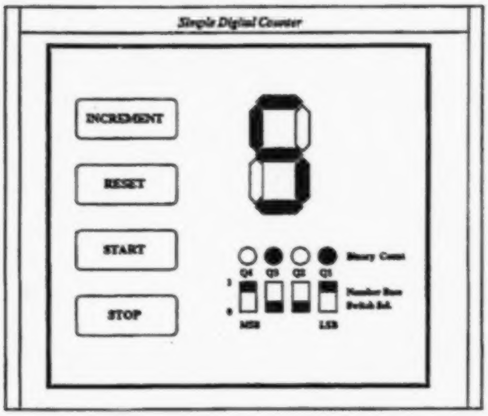


Figure 1: Simple Digital Counter Interface

The INCREMENT button increments the count whenever it is pressed and the RESET button sets the count to zero. The START and STOP buttons are used for continuous counting as in a stopwatch. Since the counter system is implemented using digital electronics all operations are performed in binary. The seven segment display should display the equivalence of the binary counter output if all the modules of the counter system are functioning correctly. A set of dip switches on the bottom right hand corner set the number base, in binary, in which the count takes place. The current setting on the interface is 1001 which corresponds to 9. Thus the counter output will increment 0, 1, 2, 3, 4, 5, 6, 7, 8 and back to 0. The counter uses 4 bits thus it can count up to 15 which corresponds to F in hexadecimal notation.

The digital counter application is implemented using a 4 bit binary counter mechanism, a seven segment display driver, a reset mechanism and the seven segment display. A block diagram representation of the digital counter application is shown in figure 2. This diagram illustrates the functionalities of the modules of the counter system and thus helps to focus on one or more modules during troubleshooting. For example, if the binary counter output is correct but the seven segment display is incorrect then the error should be in the seven segment display driver.

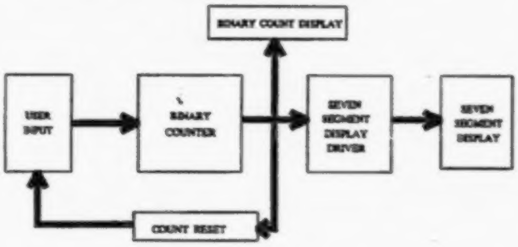


Figure 2: Functional Representation of Counter

The binary counter is implemented using a 4-stage D flip flop arrangement. The Set input of the flip flops are set to 0, enabling counting and resetting. The Reset input to the D flip flops is activated either as a result of the user pressing the RESET button or the binary count reaching the limit specified in the number base dip switches. The clock input is activated with a single pulse when the user presses the INCREMENT button or continuous pulses when the START button is pressed. The Q outputs of the D flip flop form the output of the four bit counter. The main characteristic of the D flip flop is that on the rising edge of the clock, the D input is transferred to the Q output provided Set and Reset are 0. The $\overline{Q}$ output of the D flip flop is connected to both clock of the next flip flop and the D input of the same flip flop. Each of the Q output will toggle at the rising edge of the clock connected to it.

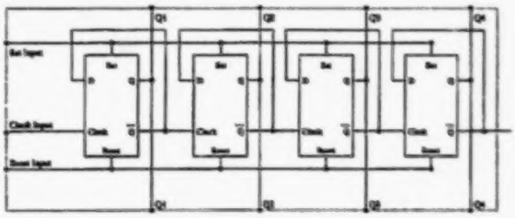


Figure 3: Four Bit Binary Counter

The reset mechanism shown in figure 4 is implemented using combinatorial logic. If the counter output and the settings on the switch are the same then the counter is reset. When this condition occurs the outputs of all the **ex-or** gates are zero and hence all the inputs to the **and** gates are one producing a reset of 1.

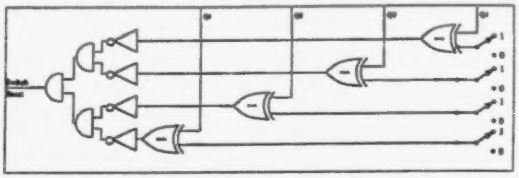


Figure 4: Dip Switch Reseting Circuitry

The seven segment driver is also implemented using combinatorial logic instead of using an off the shelf digital integrated circuit. A truth table for each of the seven segments is written down corresponding to all of the counter outputs from 0000 to 1111. Karnaugh maps are used to simplify the logic function for each of the segments. The resulting function is implemented using an and-or gate combination. It could have been implemented using a nand-nand configuration which presumes knowledge of De Morgans laws. Thus each of the seven logic functions is implemented using one and-or configuration of logic gates.

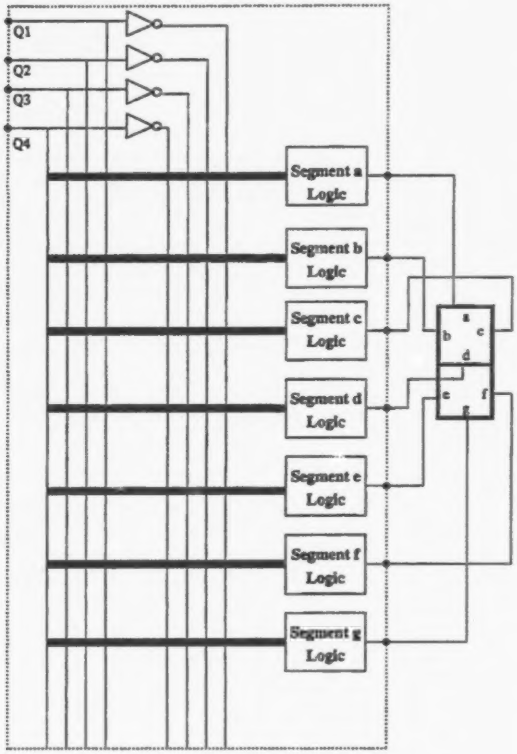


Figure 5: Seven Segment Display Driver

Models of Digital Electronics Knowledge

In our approach to fault diagnosis in digital circuits we have represented the concepts and objects associated with digital electronics in a series of increasingly complex models. At the beginning, student knowledge is represented as a set of simple models of the domain and task. During the course of troubleshooting practice and instruction, the student is introduced to more complex models of digital circuits and troubleshooting. This idea is similar to mental models used by White et al. [20]. A mental model is knowledge structure that incorporates declarative and procedural knowledge along with a control structure that combines them. Learning is viewed as a transformation of mental models, each of which is adequate for a subset of problems.

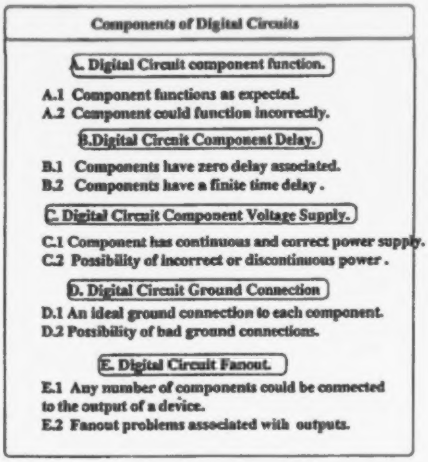
In this section we describe the models used in our work on the domain of digital electronics and the task of troubleshooting digital circuits. A digital circuit consists of a set of components and interconnections. The junctions of nodes and interconnections are referred to as nodes. Signals flow along the interconnections from the output of one component into the input of one or more components. The behavior of a circuit can be ascertained by observing the signals at various nodes of the circuit.

Digital Circuit Signals

1. **Abstract logic levels of 0 and 1.**
2. **Logic levels of 0,1 and undefined and associated threshold values.**
3. **Actual voltage values.**

Figure 6: Models of Digital Circuit Signals

The simplest model of signals in a digital circuit is the notion that they can be represented as 0's(low) and 1's(high) only. This model is adequate to solve simple faults in digital circuits. But it is inadequate to solve more complex problems within a circuit. A more complex model of digital circuit signals assigns threshold voltage levels for 0 and 1. The concept of undefined voltage which lies in between the two thresholds is introduced in this model. The second model is adequate to solve a larger subset of problems. However in real life digital troubleshooting, the abstract concepts of 0 and 1 are replaced by actual voltages which can be abstracted to zero and one. Thus the models mentioned above represent a progression of models of signals in digital circuits. The models used are shown in figure 6.



Components of Digital Circuits

A. Digital Circuit component function.
A.1 Component functions as expected.
A.2 Component could function incorrectly.

B. Digital Circuit Component Delay.
B.1 Components have zero delay associated.
B.2 Components have a finite time delay.

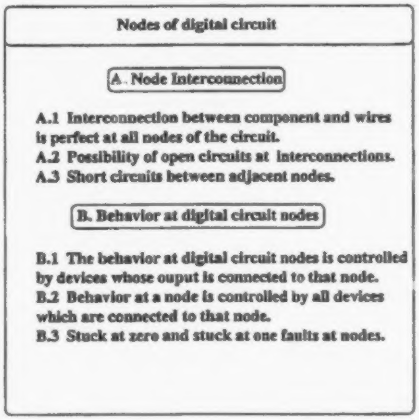
C. Digital Circuit Component Voltage Supply.
C.1 Component has continuous and correct power supply.
C.2 Possibility of incorrect or discontinuous power.

D. Digital Circuit Ground Connection
D.1 An ideal ground connection to each component.
D.2 Possibility of bad ground connections.

E. Digital Circuit Fanout.
E.1 Any number of components could be connected to the output of a device.
E.2 Fanout problems associated with outputs.

Figure 7: Models of Digital Circuit Components

A Digital Circuit Component has a number of input and output ports. It performs an operation on the input signals to produce an output. This is referred to as the function of the component. A number of concepts are associated with a digital circuit component such as the component delay, fanout, power or voltage supply, grounding, etc. The simplest model of any digital circuit component excludes all these details and considers only the input and output signals of a digital circuit component and its function. However in real-life troubleshooting all of the other concepts are very important for successful fault diagnosis. The progression of models of digital circuit components are shown in figure 7.



Nodes of digital circuit

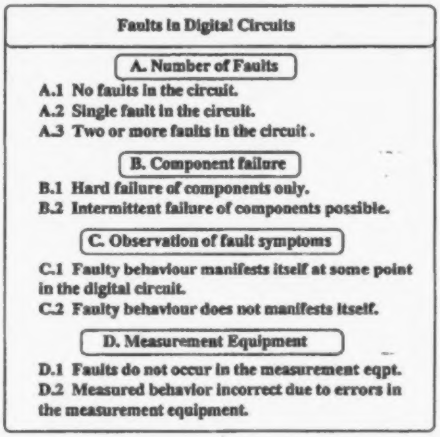
A. Node Interconnection
A.1 Interconnection between component and wires is perfect at all nodes of the circuit.
A.2 Possibility of open circuits at interconnections.
A.3 Short circuits between adjacent nodes.

B. Behavior at digital circuit nodes
B.1 The behavior at digital circuit nodes is controlled by devices whose ouput is connected to that node.
B.2 Behavior at a node is controlled by all devices which are connected to that node.
B.3 Stuck at zero and stuck at one faults at nodes.

Figure 8: Models of Nodes

Interconnection points in digital circuits are referred to as nodes. Signal measurements are usually done at nodes of digital circuits. The simplest model of digital circuit operation assumes that all nodes in the circuit are ideal ie., there is good contact between wires and components. A further assumption that is made because of the directional nature of digital devices is that the behavior at a node in a digital circuit is controlled only by the component whose output is connected to that node. However more complex models retract these assumptions as shown in figure 8.

Faults in digital circuits can be represented as a set of increasingly complex models as shown in figure 9. The student is progressed into a more complex model only if he demonstrates sufficient competence in troubleshooting faults associated with a particular model. At the beginning the student is presented with a digital circuit containing a singleton hard fault that is measurable at some point in the circuit using fault free equipment.



Faults in Digital Circuits

A. Number of Faults
A.1 No faults in the circuit.
A.2 Single fault in the circuit.
A.3 Two or more faults in the circuit.

B. Component failure
B.1 Hard failure of components only.
B.2 Intermittent failure of components possible.

C. Observation of fault symptoms
C.1 Faulty behaviour manifests itself at some point in the digital circuit.
C.2 Faulty behaviour does not manifests itself.

D. Measurement Equipment
D.1 Faults do not occur in the measurement eqpt.
D.2 Measured behavior incorrect due to errors in the measurement equipment.

Figure 9: Models of Faults

Troubleshooting Methodology

The troubleshooting methodology utilizes a wide variety of information as shown in figure 10.

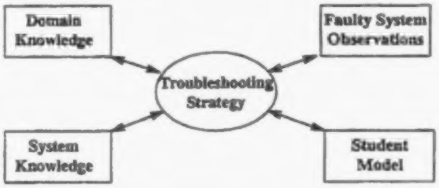


Figure 10: Troubleshooting Knowledge

The troubleshooting strategy is the procedure used to detect the fault in the circuit. The troubleshooting strategy uses digital electronics domain knowledge, digital system or application knowledge, observations of the faulty system and a model of the student knowledge. The domain knowledge captures the concepts of the digital electronics domain. This knowledge is application independent and could be used for troubleshooting any digital circuit. Information pertaining to the circuit used for instruction is captured in the application circuit knowledge. This information pertains to the circuit under normal operating conditions. Faulty system observation contains knowledge of the faulty circuit. The Student Model contains the current state of student knowledge which dictates the types of knowledge and strategy that can be used for troubleshooting.

Domain Knowledge

Domain knowledge contains the concepts of the digital electronics captured in frame-based and rule-based formalisms in the expert system. Figure 11 shows the different types of domain knowledge used during troubleshooting.

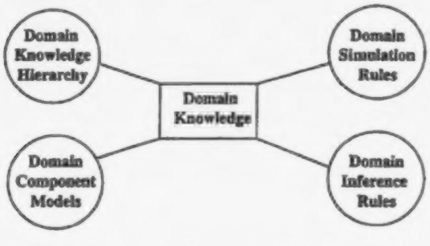


Figure 11: Domain Knowledge for Troubleshooting

A hierarchical description of the digital electronics domain captures the properties of digital circuit components and subsystems. Domain simulation rules [6, 18] use known component inputs to simulate the expected output, based on component function. Domain inference rules [6, 18] use known outputs of a component, together with component function to infer the input. Domain component models capture the input-output relationships of digital circuit components in the form of truth tables.

Application Circuit Knowledge

The digital circuit used in the ILE has to be represented in such a way that a wide variety of information about it can be extracted during troubleshooting. In our representation we have used structural, behavioral, functional and physical description of the digital system as shown in figure 12.

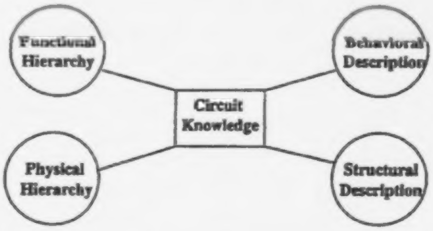


Figure 12: Circuit Knowledge for Troubleshooting

Structural information or structure refers to the components, component behaviors, and the connection between components [2]. Structural information is captured in the circuit schematic in real-life. In the expert system connectivity information is represented as facts in the data base. Components are instances of standard classes and/or objects that exist in the domain knowledge hierarchy thus inheriting the properties of the classes and objects to which they are connected to in the hierarchy. Behavior refers to changes of state of the system with respect to time due to the interactions of the behaviors of its components through specified connections. In our work we use static, time domain behavioral representation of the digital system. Function is the relation between a goal of a physical system and its behavior [7]. The functional hierarchy is a description of the digital system from a functional perspective as shown in figure 13. The root node contains the name of the digital system. The first level nodes contain the modules of the digital system. Each module performs a specific function.

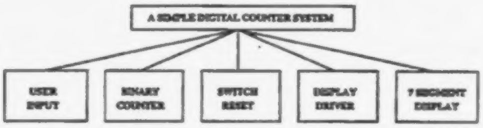
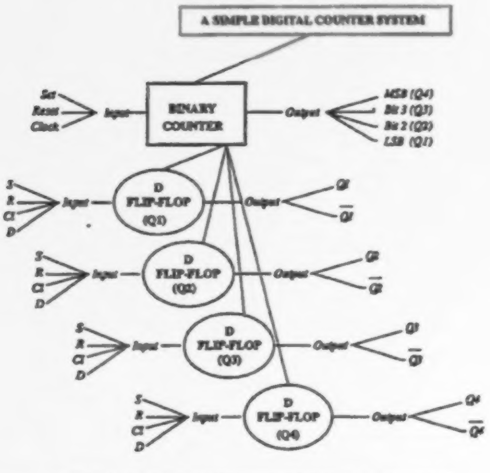


Figure 13: Functional Hierarchy of the Application

Each of the modules of the digital system can be further divided into submodules each with a distinct function. The submodules of the binary counter are shown in figure 14. All other modules of the digital system can be expanded similarly. The functional hierarchy forms the core and starting point for fault diagnosis.

Troubleshooting Strategy

The troubleshooting strategy is the procedure used by the expert system in detecting the fault in the digital system. The faulty component or module in any digital circuit can be identified through functional and structural testing of each component in the circuit.



Each of D flip-flop objects inherit properties from the domain knowledge as well.

Figure 14: Binary Counter Module

Functional tests examine the behavior of the circuit component or module over the range of possible inputs. Structural tests confirm the connectivity data of the component or module. However thorough testing of each component in a digital circuit can be an expensive exercise and has little instructional value to novices. Thus we seek to use a strategy that can simplify the troubleshooting process. Reiter [16] states " ... diagnosis need not be unique; there may be several competing explanations for the same faulty system.. "

One characteristic of troubleshooting strategy of experts [15] is that fault diagnosis proceeds in a top-down fashion in a functional hierarchy. They begin by identifying a subsystem malfunction that could account for the symptoms. Thereafter diagnosis proceeds down the hierarchy until a component is found at the lowest level of the functional hierarchy that could explain the observations.

The troubleshooting strategy we have adopted uses the functional hierarchy of the digital system to determine the circuit module or submodule that is behaving incorrectly. Thereafter the troubleshooting switches to component level interactions. The strategy is captured in the following procedures.

1) Start at the top level of the functional hierarchy.

2) Test the input-output behavior of the digital circuit.

3) Move one level down the functional hierarchy and perform a breadth first search to identify the module closest to the system input which is faulty.

4) Repeat step 3 on the faulty module or submodule until the lowest level of the functional hierarchy at which components do not appear. This corresponds to the smallest functional block whose input-output behavior is incorrect.

5) Mark this point in the functional hierarchy at which functional hierarchy is halted and switch to component level troubleshooting.

The above procedure identifies the smallest functional block closest to input which is misbehaving. The output of the functional block is isolated and component level troubleshooting begins from the output of this block. It involves working backwards from this point to identify the faulty component that is closest to the block output. The input-output behavior of this component is tested. If unsatisfactory, mark the component as suspicious and proceed to test the component after isolating it. If the input-output behavior of the component is satisfactory then compare the expected and observed behavior at the nodes corresponding to component inputs. If different, mark all components and nodes within the module that are structurally connected or which functionally control the logic level at the input node as suspicious. Members of this set are not tested in subsequent troubleshooting actions. When the size of the set of suspicious circuit elements have been reduced to a small number each component is isolated and individually tested. During subsequent tests if the behavior at some other node is incorrect ie., not expected, then all components which functionally control or structurally connected to this node are marked as suspicious. The comparison of the two sets reduces the size of the set of suspicious circuit elements.

Conclusion

We have described a methodology for troubleshooting digital circuits in an interactive learning environment. The troubleshooting principles and the knowledge representation of the digital electronics domain developed here will enable us to present simple combinatorial and sequential digital circuit applications within the interactive learning environment.

Further work would involve development and enhancement of the knowledge representation techniques and troubleshooting strategy to handle more complex circuits. The troubleshooting strategy described in this paper uses static testing to determine faults in a digital circuit. We would also like to implement dynamic testing in the future. This is because certain types of faults cannot be detected using static testing alone.

References

[1] Bloom, B.S., "The 2 Sigma Problem: The Search for Methods of Group Instruction as Effective as

One-to-one Tutoring", *Educational Researcher*, Vol 13, 1984, pp. 3-16.

[2] Bobrow, D.G., "Qualitative reasoning about physical systems: An introduction",*Artificial Intelligence*, Vol 24, 1984, pp 1 - 5.

[3] Brown, J.S., Burton, R.R., & deKleer, J., "Pedagogical, natural language and knowledge engineering techniques in SOPHIE I II & III", *IntelligentTutoring Systems* edited by Sleeman, D., & Brown, J.S., Academic Press, 1982.

[4] Clancey, W.J., *Knowledge-based tutoring: The GUIDON program.*,Cambridge, MA, The MIT Press, 1987.

[5] Chu, R.W., " Towards the tutor/aid paradigm: Design of intelligent tutoring for operators of supervisory control systems.", Doctoral Thesis, Centre for Human-Machine Systems Research, School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA, 1991.

[6] Davis, R., "Diagnostic reasoning based on structure and behavior",*Artificial Intelligence*, Vol 24, 1984, pp 347- 409.

[7] de Kleer, J., & Brown, J.S., " A qualitative physics based on confluences", *Artificial Intelligence*, Vol 24, 1984, pp. 7- 83.

[8] Edwards, M., "The Mercedes Benz of interactive video" *Hardcopy*, Vol. 14, No. 5, pp. 74-80, May 1985.

[9] Fath, J.L., Mithcell.C.M., "An ICAI Architecture for Troubleshooting in Complex Dynamic systems", *IEEE Transactions on System, Man and Cybernetics*, Vol SMC-20, No.3, May/June 1990.

[10] Fink, P.K., & Lusth, J.C. "Expert Systems and Diagnostic Expertise in the Mechanical and Electrical Domains", *IEEE Transactions on System, Man and Cybernetics*, Vol SMC-17, No.3, May/June 1987.

[11] Fink, P.K., "The Role of Domain Knowledge in the Design of an Intelligent Tutoring System",in *Intelligent Tutoring Systems: Evolutions in Design* ed by Hugh Burns, James W.Parlett & Carol Luckhardt Redfield, Hillsdale, N.J.: L. Erlbaum, 1991.

[12] Govindaraj, T., & Su, Yuan-Liang D. " Model of fault diagnosis performance of expert marine engineers.", *International Journal of Man-Machine Studies*, v 29, n 1, Jul 1988, p 1-20.

[13] Lesgold, A., LaJoie, S., Bunzo, M., & Eggan, C., "SHERLOCK: A Coached Practice Environment for an Electronics Troubleshooting Job", edited by Chabay & Larkin, *Computer Assisted instruction and Intelligent Tutoring Systems: Establishing Communication and Collaboration*, Lawrence Erlbaum Associates, NJ, 1991.

[14] Mitchell, C.M., & Govindraj, T., " A tutor for satellite system operators", *Proceeding of the 1989 IEEE International Conference on Systems, Man and Cybernetics"*, Nov 14-19, Cambridge, MA, p 766- 770.

[15] Rasmussen.J., "Information processing and human-machine interaction: An approach to cognitive engineering", New York, North-Holland, 1985.

[16] Reiter, R., "A Theory of Diagnosis from First Principle", *Artificial Intelligence*, Vol 24, 1985, pp 57 - 95.

[17] Rickel, J.W., "Intelligent Computer-Aided Instruction: A survey organized around system components", *IEEE Transactions on systems, man and cybernetics*, Vol 19, No 1, January/February 1989, pp. 40-57.

[18] Tzafestas, S.G., " System fault diagnosis using the knowledge-based methodology", in *Fault Diagnosis in dynamic system theory and applications*, ed Patton.R, Frank.P & Clark.R, Prentice Hall, 1989.

[19] Vasandani, V., & Govindraj, T., " Knowledge Structures for a computer-based training aid for troubleshooting a complex system", to appear in *The Use of Computer Models* ed by D.M.Towne, Ch 15, Nato ASI Series F, 1993.

[20] White, B.Y., & Frederiksen, J.R., " Causal Model Progression as a Foundation for Intelligent Learning Environments", *Artificial Intelligence*, Vol 42, 1990, pp. 99-157.

909957

A Simulation-Based Approach to Spacecraft Power System Fault Detection and Diagnosis

Peter Adamovits[1]
Canadian Space Agency
Communications Research Centre
3701 Carling Avenue
Ottawa, Ontario, Canada, K2H 8S2

#149685

Abstract

The paper describes the model-based Multiple Fault Diagnostic System (MFDS) that identifies and diagnoses faults that occur in a spacecraft Electrical Power System (EPS). The spacecraft on which the MFDS is demonstrated is large and complex. Communication opportunities are infrequent and short in duration. Effective management of the spacecraft is difficult for the human operators. High costs may result from operator error including loss of the spacecraft. The demand for increased reliability has resulted in the push toward increased automation. The MFDS attempts to assist the operator by monitoring the EPS identifying degraded or failed components.

In knowledge based systems, there is a distinct division between the knowledge and the mechanisms that operate on the knowledge. Here the knowledge component is further partitioned into the deep behavioral knowledge of the domain (the model) and the shallower knowledge of the reasoning elements. The reasoning elements are divided into 6 discrete steps, the Diagnostic Process. At each diagnostic process step, a shallow reasoning system is used to reason about the data, symptom or hypothesis set under study. The partitioned knowledge base is easier to create and maintain than the flat knowledge base that typically exists.

The model based reasoning approach used in the MFDS relies on a simulation for the model. Here the engineering details that system designers use to construct their simulation, test its accuracy and reliability in a variety of operational modes are also used in the diagnostic system model. The reliance on a simulation eliminates the time and cost associated with recoding the model in some other form (logic, rules, etc.).

The use of a simulation as the model, shallow reasoning elements and the diagnostic process has proven successful. The Multiple Fault Diagnostic System has successfully demonstrated the approach detecting and diagnosing several tens of faults at the same time with a high accuracy.

Introduction

Model-Based Artificial Intelligence (AI) software approaches to diagnosis require encoding a reasonable facsimile of the problem domain. This model is used as a reference for comparison with the data received from the domain being examined. Often, the model is distributed throughout the knowledge base violating one of Artificial Intelligence's tenets, that of the separation of the knowledge from the mechanism that manipulates the knowledge. In the work presented, the model is encoded as an object oriented simulation (continuous, linear) of the domain, a spacecraft electrical power system. The inferential elements of the system are kept independent and separate from the model, the fore mentioned simulation.

In this approach the fault detection and portions of the diagnostic reasoning system thus become comparitors between the expected behavior (that of the model) and the application domain under analysis. The diagnostic problem is partitioned into 6 discrete steps that include: fault detection, diagnosis and hypothesis generation, hypothesis space pruning, validation of the hypothesis, test case selection and verification of the diagnosis. The system performs a simplified form of learning by injecting known diagnosed faults into the model (compilation) to keep it consistent with the domain reducing further unnecessary processing in the future. It has been demonstrated here that largely shallow reasoning systems performing a comparative analysis of the data is sufficient for a complex application.

---

[1] now at Bell Northern Research, P.O. Box 3511, Station C, Ottawa, Ontario, Canada, K1Y 4H7

The approach has been successfully demonstrated on a software simulation of the Space Based Radar Electrical Power System (EPS) breadboard now under development at the Canadian Department of National Defence. In terms of model based reasoning systems previously developed, the application domain is complex with more than 15,000 components and 35,000 sensor points.

The Application Domain

The US Air Force (USAF) with the Canadian Department of National Defence (DND) are together examining techniques for space based surveillance of North American airspace. That is to detect, localize and track aircraft from space. The Space Based Radar (SBR) program is intended to meet these needs through the use of a high power radar payload spacecraft at a highly inclined, low earth orbit. The Canadian SBR Research and Development (R&D) project will improve Canadian industries' understanding of the SBR concept and improve the technology base within Canada, so that Canada may contribute significantly to a collaborative SBR program [Eatock 1990]. As part of Canada's contribution, a hardware breadboard of the SBR spacecraft's Electrical Power System was designed and built by a team from CAL Corporation and Spar Aerospace [Moodie and Maskell 1989].

Space Based Radar-Electrical Power System

The SBR spacecraft is expected to have a total electrical power demand on the order of 40K watts. To reduce the power carrying capacity requirements for the individual electrical components and increase reliability, the EPS is divided into 10 parallel EPS strings. Each string is identical and provides 1/10th of the power requirements of the spacecraft.

The EPS breadboard now under development implements one EPS string and includes the EPS elements: solar panels, batteries, power distribution conductors, power conditioning and regulation components and loads. The breadboard also includes control electronics and computing elements to manage and control the EPS. Each string includes some 1548 electrical components (Fig.1) each of which has a variety of failure modes. The health and effectiveness of the EPS and health of the components is monitored through sensors located at points of interest. There are some 3500 sensors that measure voltage, current, temperature, power (electrical or RF) and logical (status) values in the base line breadboard design of one SBR-EPS string. The sensor values are available from the breadboard *on demand* and are available through a simulated telemetry stream. (It is expected that sufficient bandwidth will not be available on the telemetry channel to transmit this data continuously.)

While the breadboard EPS design has some level of fault isolation through circuit breakers and fuses, the system does not reason about the causality of the symptom nor the specific component that caused the symptom to appear. The Canadian Space Agency (CSA) Research Branch has considerable experience in developing Knowledge Based Reasoning Systems to address that address fault detection, diagnosis and scheduling / planning. As part of ongoing research in *model based reasoning* as applied to fault detection and diagnosis the CSA has been working cooperatively with DND, CAL and Spar to develop a system to address the SBR-EPS.

The Multiple Fault Diagnostic System (MFDS) described in this paper is intended to assist the spacecraft operator in monitoring the spacecraft EPS operation, detect the existence of faults, identify the component causing the fault and if possible prescribe actions that will either provide an alternate configuration or isolate the fault.

Model Based Diagnosis

Model based reasoning has generated considerable interest in the research community particularly as applied to diagnosis. Systems have been developed by several noted researchers [Davis 1984; Genesereth 1984; deKleer and Williams 1987; Hamscher 1991 and Reiter 1987].

In model based reasoning systems, the model is used as a reference to which the observed behavior of the domain system is compared. These models are based on theories or experimental data from the domain. *The model is not based on experiential data from the domain but a deeper understanding.* To be effective, the model must accurately describe the underlying *science* of the domain. Models may be constructed (encoded) in a variety of forms (logic, frames, rules, procedural code and others). The models operate either in a quantitative or qualitative domain [Kuippers 1986]. Rothenberg [Rothenberg 1989] (page 75) defines modeling in the following general terms:

> *"Modeling in its broadest sense is the cost effective use of something in place of something else for some cognitive purpose. It allows us to use something that is simpler, safer, cheaper than reality, instead of reality for some purpose."*

Modeling or model based reasoning however are not sufficient to ensure the above results. Brittleness of the reasoning process is the breakdown of the reasoning process due to: a lack of sufficient understanding of the underlying mechanisms that exist in the domain; the use of an inappropriate or insufficiently expressive language, tool or technique and others. It is the belief of this author that through the use of object oriented modeling and programming techniques significant gains can be realized.

Object oriented techniques are ideally suited to modeling elements of a system where the boundary values and describing equations or relationships of the component are well known. For example: The SBR-EPS contains several power regulation and conversion components with known operational voltages, currents and operational efficiencies.

For the solar panel the first order behavioral model is based on a piece wise continuos equation that describes the voltage / current characteristic of the photo voltaic arrays:

voltage volts	current amps
170	0
160	25
0	30

Describing equations:
voltage = 170 - ((10/25)*current) for 0<current<25
voltage = 160 - ((160/5)*current) for 25<current<30

The above relationship is in fact encoded in a Smalltalk class that defines the voltage-current characteristics of solar panel in the model. It should be noted that the granularity of the model is limited by the level that sensor data is available from the EPS. The EPS does not contain sensors at individual cells or groups of cells in the solar panel. Diagnosis a level below the complete solar panel is thus not possible. Modeling at this lower level will have little or no benefit in this case.

The EPS model is constructed by connecting instances of models of the individual components that comprise the EPS. While each string of the EPS contains some 1548 component, there exist only 9 unique components within the EPS. These are: solar panel, primary power conditioner, battery cells, cell packs and battery assembly, intermediate conditioners, transmit receive conditioners, auxiliary conditioners and radar loads. Once the connectivity of the components has been established, the various components of the EPS model are updated by performing a power (energy) balance across the network. This is accomplished by propagating the power demand from the loads through the intermediate components to the source components and then performing a power balance (using Kirchoffs laws) at the source components (solar panel and batteries). (The above discussion was based on material drawn from Adamovits [Adamovits 1992]).

The Multiple Fault Diagnostic System

The Multiple Fault Diagnostic System (MFDS) is a model based reasoning system that detects diagnoses and verifies EPS components' faults. [Adamovits 1992, Adamovits and Pagurek 1993] It is capable of identifying multiple concurrent faults that may have either interacting or independent symptoms as reflected in the sensory data. The MFDS relies on a model of the EPS that is in the form of a software simulation containing both continuous and discrete elements. The majority of the MFDS was implemented in Smalltalk [Digitalk 1988] an object oriented language. The Prolog language [Clocksin and Mellish 1982] with a few changes to control the backtracking was used as a simple rule base interpreter for the inferential reasoning components of the MFDS. All rules used in this system are shallow in nature. That is, the rules do not contain a deep understanding of the underlying behavior of the components of the EPS but instead have a shallow understanding of the differences between the expected and observed behavior. These shallow rules bases conclude possible hypotheses based on the observed behavior.

While the rule bases contain largely shallow rule structures, the MFDS is capable of addressing the multiple fault application through the use of the deep knowledge contained within the model.

Diagnostic process

In the Multiple Fault Diagnostic System (MFDS) the fault detection/diagnosis through to hypothesis testing on the spacecraft EPS is broken down into several discrete steps. These are referred to here as the *Diagnostic Process*. The Diagnostic Process is the activity of identifying the underlying causes for unexpected changes in a system's behavior. It includes several key reasoning steps that must be completed along the way. For the purposes of this paper the Diagnostic Process is defined to include the steps:

- data acquisition,
- fault detection,
- diagnosis and hypothesis generation,
- hypothesis space pruning,
- hypothesis validation (testing) on a model of the system,
- test case selection,
- hypothesis verification (testing) on the EPS.

The Diagnostic Process is illustrated in Fig.2 and is further described in the following sections paragraphs [Adamovits and Pagurek 1993]:

Data acquisition - The MFDS implements the data acquisition system as a data buffer that samples and holds the data resulting from the model and the spacecraft EPS[2]. These data sets are maintained until a new sample command is issued. The sample and hold approach ensures that the data sets remain consistent through all remaining steps of the diagnostic process. (This approach is also consistent with the expected mode of operation of the spacecraft as sensor data is only expected to be available on request.)

Fault detection - Present day space industry approaches to fault detection typically involve comparing the received data against fixed (with respect to time) upper and lower set points. That is limit checking. A fault is detected when the telemetry data deviates from the range defined by the limits specified. These limits are typically set manually on an infrequent basis by the spacecraft operator. On occasions when transients are expected to occur, the limits are set quite wide to reduce the possibility of false alarms. Widely set limits however serve little no value in monitoring the rates of change in the data. This can only be accomplished by other means. Fault detection performed in this way is effective for largely static data only.

The SBR-EPS experiences high power transients as the radar payload switches on and off. In this dynamic environment, static limits are ineffective. To meet the needs of the SBR-EPS, fault detection is accomplished through two parallel reasoning components performing qualitative and quantitative fault detection respectively. In both cases these components perform a comparison between the expected data from the model with that received from the domain.

The quantitative fault detection component performs this as a numeric (or logical) comparison between the two data sources (domain and model). If the difference exceeds some threshold, a fault is postulated. This is analogous to the limit checking approach of the typical spacecraft operations center. In this case however, the simulation capability of the model presents state varying data that is used to establish state varying limits. The range of these limits upper and lower limits is established to suit the individual data element (figure 3).

Qualitative fault detection is accomplished by qualitatively interpreting the quantitative data from both the domain and the model. The qualitative values (decreasing, no change and increasing; {- 0 +}) are then compared to determine if the *direction of change* of the domain data is that predicted by the model. If not, a qualitative fault is indicated. (The pairs {(- -) (0 0) (+ +)} indicate no qualitative fault. All others indicate a fault that is dependent on the context of the EPS and the component being examined). This qualitative fault detection approach is similar to the qualitative reasoning systems of Kuippers [Kuippers 1986] and others.

The fault detection system (qualitative and quantitative fault detection components combined) returns a set of plausible faults based on a shallow analysis of the available data. That is, only a symptom based analysis of comparing the sensor data from the model and domain is performed. This approach however, includes the deep knowledge aspects of the domain by including the dynamically changing behavior of the EPS model in the symptom based analysis.

Diagnosis and hypothesis generation - Based on the results of the fault detection system, the diagnostic system identifies combinations of symptoms (detected faults) that indicate specific component failures.

---

[2] During the development of the MFDS the SBR-EPS breadboard was not complete. To provide representative spacecraft data, another instance of the model was created and used to represent the spacecraft.

Many of the faults identified by the fault detection system are in fact the propagation of symptoms from lower level components. For example:

> If a component (e.g. load 1) is drawing more power than anticipated then, the power source components (e.g. IntCond22 and TRCond1) for this failed component will also be likely drawing more power than predicted by the model. A failure however does not exist in these other components.

The diagnosis and hypothesis generation system attempts to reduce the number of incorrectly detected faults and prevent their further propagation to later steps in the diagnostic process. While the analysis here is also shallow in nature, a certain degree of deep knowledge is also contained within the diagnosis and hypothesis generation knowledge base. This deep knowledge is in the form of knowledge of component connectivity and knowledge of how components may pass symptoms when they are not in fact at fault.

The output of the diagnosis and hypothesis generation system is a set of plausible diagnosis hypotheses. Each plausible hypotheses is supported by some subset of the detected faults or sensor and model data from the data interface system, *the hypothesis is locally consistent with the data.*

A few poorly supported and incorrect diagnoses will remain however. By poorly supported it is implied that the hypothesis is inconsistent with other hypotheses that have been identified. That is: *the poorly supported hypotheses are globally inconsistent with each other when a deep analysis is performed.* The removal of inconsistent or poorly supported hypotheses is performed in the next step in the diagnostic process.

Hypothesis space pruning - The set of hypotheses resulting from the hypothesis generation step are examined discarding those that are not strongly supported or are contrary to the expected based on other believed hypotheses. Hypotheses may be rejected based on a variety of criteria (known operational state of the spacecraft, solar conditions, etc.).

The MFDS implements a hypothesis space pruning step that is based on the knowledge of how symptoms propagate from electrical source components to lower level electrical components. For example, if the set of hypotheses from the diagnosis and hypothesis generation system includes the hypotheses {... (load1 to load4 open circuit) ... (transmit receive conditioner open circuit)...} then we may assume that if these components are related electrically, then the load1 to load4 open circuit hypothesis is poorly supported. This is based on the fact that if the load has no source electrical power then it can expect to also have the hypothesis of open circuit. The load open circuit hypotheses are thus further symptoms of the higher level (electrically a source component) component's fault.

The result of the pruning step is a *minimal ordered list* of plausible supported hypotheses. These become the basis of the hypothesis validation step.

Hypothesis validation (with the model) - In an effort to diagnose the fault and resolve the problem, several competing hypotheses may remain on the pruned hypothesis list. The most likely of these is selected for further detailed examination.

The MFDS performs the hypothesis validation step through the use of the EPS model before proceeding. First, the most likely hypothesis is selected from the set of pruned and ordered hypotheses. To verify this candidate, the *model* is first altered to represent the fault identified in the hypothesis (e.g. if a component is hypothesized to have an open circuit, then the component model will be changed to represent the open circuit). At this point, the data acquisition, fault detection, diagnosis and hypothesis generation and hypothesis space pruning steps are repeated.

During the process of propagating the revised model data through the reasoning systems, a meta-level reasoning system monitors the progress. If following the examination of the new model data, the selected hypothesis is no longer in the set of pruned and ordered hypotheses, then the selected hypothesis is a plausible representation of the identified fault. At this point the selected hypothesis can be considered to be validated. If this is not the case, the hypothesis will be rejected and another hypothesis will be selected by the meta-level reasoner and the process is repeated. This will be continued until at least one valid hypothesis can be identified.

It should be noted that the identification of a plausible hypothesis is not sufficient. Several fault hypotheses may identify identical symptoms (e.g. a tripped circuit breaker and an open circuit within the same component). The resolution of these can often only be performed by further testing in the domain (e.g. resetting the circuit breaker). This further step is completed during hypothesis verification.

Test case selection - Once a hypothesis has been validated and identified as a plausible representation of the problem situation, a test is selected to provide either further supporting or refuting evidence when applied in the domain. In situations where component degradation and failure modes are known, these tests can be recorded and stored in a library of accepted cases. In situations that are new and unanticipated, a test case must be devised as necessary.

The test case selection system of the MFDS uses the verified hypothesis as an index into the case library of generic component test cases. The selected case is altered to include information of the component identified by the validated hypothesis. The test case is applied in the next step in the diagnostic process.

Hypothesis Verification (testing on the domain system) - Once a hypothesis has been selected and validated the spacecraft configuration is modified through the application of the test scenario. The test case may require resetting circuit breakers, turning components on or off etc. The intent is to either resolve the problem (fault recovery) or to provide further supporting or refuting evidence to be examined in an effort to improve the belief in the hypothesis describing the fault.

Hypothesis verification is accomplished by sending commands and observing the results in the telemetry data. As in the previous steps, the MFDS performs this in a fully automated way[3]. During verification, both the model and domain are altered in *identical ways*. That is, actions to acquire additional data from the domain are also applied to the model. If the predicted hypothesis is correct then the actions performed to the model and domain should indicate some expected behavior. As part of the verification step, the data acquisition, fault detection, diagnosis and hypothesis generation and pruning steps are repeated. The results from these systems are examined to determine the success in recovering from the hypothesized fault or providing further supporting or refuting evidence of its accuracy.

Discussion

The MFDS was constructed and successfully demonstrated against a software simulated SBR-EPS[4]. In this software simulated environment the MFDS operated to the extent of the knowledge contained within the knowledge base (solar panel, primary power conditioner and battery components were not diagnosed.) The remaining components that were diagnosed represent 1540 of the 1548 components that comprise the EPS (i.e. 99.5% of the total). The inclusion of these other components was not an indication of the limitation of the approach but rather a limitation on the available resources (i.e. time).

Model based reasoning has shown great promise for diagnostic systems. However, considerable computational cost is often associated with the approach. This cost is due largely to the cost of updating the model and maintaining it consistent with the domain. In the approach demonstrated, a very efficient model was created using object oriented techniques. It is believed that through the appropriate techniques such as a object oriented simulation that model based reasoning can be very effective in the diagnosis realm.

When compared to typical diagnostic systems, the MFDS has several benefits. These are:

- The model is constructed using objects that represent physical components as building blocks. These component models describe well known behaviors of the components and are well defined.
- The model contains the majority of the deep knowledge that describes the domain
- Monolithic knowledge bases, difficult to maintain, are avoided and are replaced by the knowledge bases identified in the Diagnostic Process. The reasoning systems contain largely shallow knowledge and are very small specific in nature.
- The system learns by adapting to changes in the behavior in the domain system (in this case the SBR-EPS) by altering the model to represent the changes in operational characteristics of the components. This learning process saves future computations by accurately modeling the already recognized faults.

Conclusion

The Multiple Fault Diagnostic System has been shown to be an acceptable approach to diagnosis in large scale problem domains (1500 elements, 3500 data points). An object oriented simulation has been demonstrated to be an appropriate approach to modeling. It is expected that the simulation/model based reasoning approach and the Diagnostic Process will find application in any domain where the behavioral parameters can be modeled with some accuracy.

---

[3] During transition to full automation, the MFDS will be expected to operate in an advisory capacity only. As the diagnostic sysetm accuracy and reliability becomes better known, closed loop operation as described will likely be used.

[4] The SBR-EPS breadboard was not complete at time of testing

References

Adamovits, Peter J. 1992. "Model Based Reasoning Applied to the Diagnosis of Spacecraft Electrical Power System Faults", masters thesis, Mechanical and Aerospace Engineering, Carleton University, 175 pages.

Adamovits, Peter J. and Bernard Pagurek, 1993, "Simulation (Model) Based Fault Detection and Diagnosis of a Spacecraft Electrical Power System", 9th Conference on Artificial Intelligence Applications, IEEE, 3-5 May, Orlando, Florida.

Clocksin, W.F. and Mellish, C.S. 1982. "Programming in Prolog", Springer-Verlag, 279 pages.

Davis, Randal. 1984. "Diagnostic Reasoning Based on Structure and Behavior" Artificial Intelligence, Elsevier Science Publishers, Volume 24, Pages 347-410.

Digitalk. 1988. "Smalltalk/V 286 Tutorial and Programming Handbook", Digitalk Inc., 561 pages.

Eatock, Brian C. 1990. "Progress on DND's Space-based Radar R&D Project", Sixth CASI Conference on Astronautics, Ottawa, November, Pages 452-463.

Genesereth, Michael R. 1984. "The Use of Design Descriptions in Automated Diagnosis" Artificial Intelligence 24, pages 411-436.

Hamscher, Walter C. 1991. "Modeling Digital Circuits for Troubleshooting", Artificial Intelligence 51, Elsevier Science Publishers, Pages 223-271.

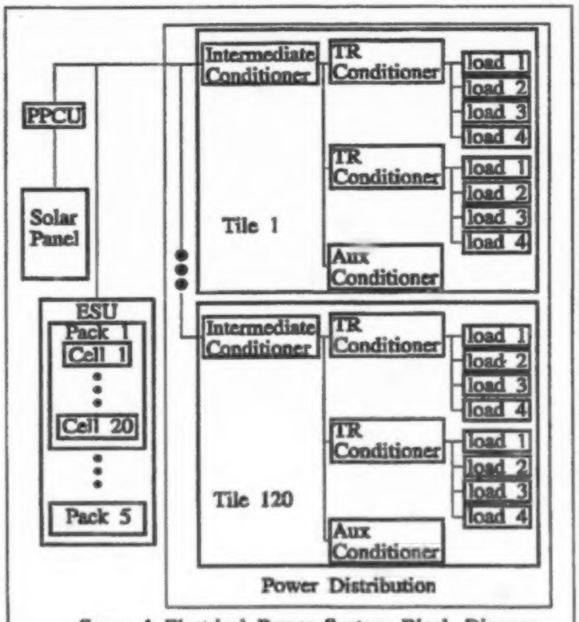
deKleer, J. and Williams B.C. 1987. "Diagnosing Multiple Faults". Artificial Intelligence 32(1), Elsevier Science Publishers, Volume 32, Pages 97-130.

Kuipers, Benjamin. 1986. "Qualitative Simulation" Artificial Intelligence Volume 29 Elsevier Science Publishers, pages 289-338.

Moodie, M.H. and Maskell, C.A. 1989. "Electrical Power Distribution on Space based Radar Satellites" IEEE AES Magazine, November.

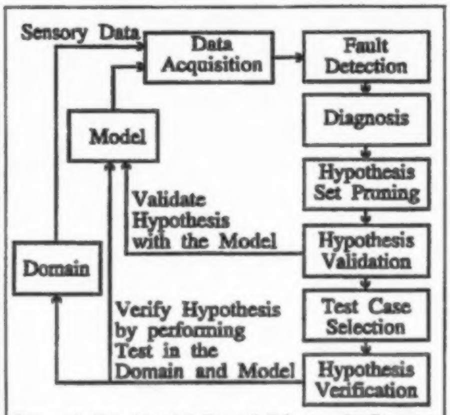
Reiter, Raymond. 1987. "A Theory of Diagnosis from First Principles", Artificial Intelligence 32(1) Elsevier Science Publishers, Pages 57-95.

Rothenberg, Jeff. 1989. "The Nature of Modelling", Artificial Intelligence, Simulation and Modeling. Widman, Lawrence E.; Loparo, Kenneth A. and Nielsen, Norman R. editors, John Wiley and Sons, pages 47-74.



PPCU
Solar Panel
ESU
Pack 1
Cell 1
Cell 20
Pack 5
Intermediate Conditioner
TR Conditioner
load 1
load 2
load 3
load 4
TR Conditioner
load 1
load 2
load 3
load 4
Aux Conditioner
Tile 1
Intermediate Conditioner
TR Conditioner
load 1
load 2
load 3
load 4
TR Conditioner
load 1
load 2
load 3
load 4
Aux Conditioner
Tile 120
Power Distribution

figure 1 Electrical Power System Block Diagram



Sensory Data
Data Acquisition
Fault Detection
Model
Diagnosis
Hypothesis Set Pruning
Validate Hypothesis with the Model
Domain
Hypothesis Validation
Verify Hypothesis by performing Test in the Domain and Model
Test Case Selection
Hypothesis Verification

figure 2 The Model Based Diagnostic Process

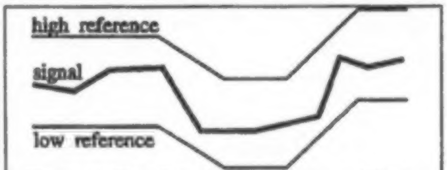


Figure 3 Signal and References vs. Time

401081

Designing Knowledge-Based Advisory Systems for Diagnostic Problems in Engineering

#149686

G. Akhras & H. C. Foo
Knowledge-Based Systems Group
Department of Civil Engineering
Royal Military College of Canada
Kingston, Ontario, Canada K7K 5L0

Abstract

Due to the combined effects of changes in service requirements and environmental conditions, engineers are confronted with increasing problems of deteriorating equipments, buildings and infrastructures needing repairs and retrofitting. Diagnosis of the type and severity of the damage is usually among the first and one of the most important stages of any maintenance program. Diagnosis process involves inspecting, assessing and correcting damage. The process lends itself well to a knowledge-based system application for two reasons : reliability and productivity. To ensure the reliability and consistence of results, knowledge-based advisory systems solve specific problems by applying appropriate rules, which have been organised into a knowledge base on the basis of standardized guidelines as well as expert opinions. Knowledge-based advisory systems are productivity tools which can be used to transfer expert knowledge to non-expert personnel, thus increasing the productivity without compromising on the reliability of the results. The main objective of this paper is to propose and discuss a new approach for designing object-oriented knowledge based advisory systems for diagnostic problems in engineering. Two such systems, which are being developed at the Royal Military College of Canada by the authors, are presented as illustrative examples.

Introduction

As a result of the increasingly high cost of new construction or production coupled with the ever-shrinking spending power, the importance of maintenance and rehabilitation of existing resources has been recognized and emphasized by all responsible agencies worldwide. Equipment or property maintenance involves diagnostic analysis and problem correction processes.

The goal of a diagnostic analysis is to ensure an equipment's or a building's performance, both structural and non-structural. Its objective is to identify and diagnose problems and to select and evaluate repair systems corresponding to the type and severity of the problems. Periodic inspection and repair of minor problems prevent the occurrence of more serious problems and make less frequent the need for major repairs to the facility.

Engineering Diagnosis

Generally speaking, there are four essential elements in an engineering diagnosis process:

1. identification and classification of the problem;
2. selection of treatment alternatives for a particular problem;
3. evaluation of resource requirements for each treatment, such as equipment, manpower and time; and
4. determination of the most cost-effective maintenance activity by comparing the equivalent costs of suitable types of treatment.

When performing a diagnostic analysis of a piece of equipment or a building or a section of a highway, the engineer requires information on how to interpret variations in structural and material defects, and information on the corrective actions that can be taken. The engineer must also know the suitable equipment and methods to correct the problem. A critical area of concern to the engineer is the determination of the nature, type and severity of the problem and the most appropriate or cost-effective treatment alternative.

Diagnostic analysis requires individuals, usually expert engineers, who are qualified to perform the analysis. These engineers should, by education, training and experience, have the ability to read and understand specifications and relevant code requirements. The engineers rely not only on standard operating procedures and specifications but also on years of experience and rules of thumb. As a result, the knowledge needed to conduct a reasonably accurate evaluation of a property under consideration is enormous and not readily transferable.

Human experts can not be expected to follow precisely logic, giving consistent, sound and reliable results from every single equipment or from the same building every time. The likelihood or inconsistent and sometimes poor judgement can be very high. There are other key features of the typical information-handling problems encountered in any diagnostic analysis such as lack of trained personnel and vacuum created due to staff shuffling. There is an increasing need for the application of knowledge-based systems to spread the well-established techniques for an efficient and reliable diagnostic analysis and for the maintenance or restoration of our resources.

Knowledge-Based Approach

Key benefits of a knowledge-based approach can be summarized in two words : productivity and reliability. Potential benefits of knowledge-based applications include the elimination or reduction of such common problems as the lack of diagnosis experts, the vacuum created during staff shuffling, the time spent in retraining staff, and the unavailability of source of reference such as standard specifications and expert advice in the field. All these benefits result in improved quantity (productivity) and quality (reliability) of an diagnostic assessment which is otherwise a complex subject.

Knowledge-based systems (KBS) are computer programs that capture human knowledge and decision making processes. There are two basic components in a KBS : a knowledge base which captures information or human knowledge, and an inference engine that interprets the knowledge base to make decisions. KBS are capable of performing various tasks such as diagnosis, planning, scheduling and design.

Today, diagnosis is one of the largest application domains of knowledge-based systems, which incidentally were first used in medical diagnosis practice about two decades ago. In medical applications, a knowledge-based system accepts patient symptoms, diagnoses the disease and proposes a course of treatment. Knowledge-based systems play similar role in engineering applications, in that maintenance alternatives are recommended on the basis of known and existing conditions of the equipment, building or structure.

Properly designed and developed systems facilitate the diagnostic process by providing the engineers with detailed guidance in the inspection and assessment processes and with a repair solution for the given description of the condition of a building or an equipment. Adaptive problem solving capacity of knowledge-based advisory systems is characterised by a decision-making process that can be structured for the specific case at hand. Basically, knowledge-based advisory systems support the decision-making process and incorporate one or more knowledge bases.

Designing KB Advisory Systems

Due to the rapid advance in computer technologies, the integration of KBS with algorithmic systems and databases and other technologies is common in practice. Many of the systems, either existing or under development, are hybrid systems where the KBS interacts with external programs and databases, or is only one component of a larger system.

Keeping that in mind, consideration should be given to three aspects, namely portability or connectivity, maintainability and potency, when developing a knowledge-based advisory system for a specific domain. Otherwise, the system will not likely be received well by the end-user and eventually remain on the research and development shelves forever. The following sections briefly describe and discuss, using two KBS (PAVE in Fig. 1 and TRUSS in Fig. 2) under development by the authors as examples, the required features that encompass the above-mentioned aspects of KBS.

Knowledge-Based System Development Shell

A knowledge-based system can be developed using high-level programming language or a development tools ("shell"). A "shell" is a software package that provides the developer with standardized inference mechanism, user interface capability, and knowledge acquisition module. The system developer needs to customize an interface and to add and structure the knowledge that is specific to particular domain. Nowadays, over 90% of the knowledge-based systems are developed using a development tool.

The selection of the development tool for any KBS is critical for the initial development as well as the final acceptance and maintenance of the system. There are many KBS development software available commercially and the selection should be based on the knowledge handling and operational characteristics of one or more development tools, which best suit the needs of the specific application. A high-end expert system tool, such as Level5 [Information Builders Inc. 1990], which integrates object- and knowledge-based programming is desirable.

Object-Oriented Programming

Most of the existing and operational knowledge-based systems are rule-based using inferencing engine such as backward chaining or forward chaining. In a conventional rule-based system, the domain knowledge is captured in the form of rules and its inference engine operates on these rules. Forward-chaining reasoning is data-driven in that it starts with known data and infers new data until no further conclusions or data can be deduced from the initial data. Backward-chaining reasoning is goal-driven in that it starts with a desired goal and proceeds backwards along a chain of reasoning in an attempt to gather the information or data needed to verify the goal.

In addition to shortcomings such as uni-directional reasoning strategy and lack of graphical interface that are inherent with rule-based systems, it is also very difficult to elicit knowledge and expertise in the form of a complete and accurate set of rules. Components such as frames and nets can also be used to construct a knowledge base. The recent advance in and movement towards object-oriented programming (OOP) will likely subsume all these approaches.

Object-oriented programming classifies and organizes data within a software application much the same way as we normally classify and organize our knowledge. In an object-oriented environment, data and programs are represented as objects which consist of behaviour, relations and values. For example, a human, being an object, has eating and talking as behavioral qualities, maintains relations with parents and brothers, and has values such as blue eyes and brown hair. Each object (human) is a member of a class (mammal) and the classes can be organized into a hierarchy.

The primary new features of object-oriented programming are encapsulation and inheritance [Cox 1987, Goldberg and Robson 1985, Pascoe 1986, Tesler 1986]. Data encapsulation, or information hiding facilitates modularity, allowing programs to be structured and constructed as discrete and reusable modules or objects. In other words, the code is attached to objects. In conventional computer programming, data is assessed and manipulated in a passive manner. In object-oriented systems, objects have the intelligence of knowing when and how to perform operations on themselves in an active manner.

A class is the general category of similar objects and an object in one class can inherit data structure and behaviour from another class through the use of inheritance. This property simplifies the process of building complex structures and speeds this process exponentially. Objects are used to represent the properties of a data structure and the operations permitted and performed on the structure. Generally speaking, an object is "the sum of its data and procedures" in such a way that the objects perform operations on themselves.

The object orientation of knowledge-based systems is particularly evident in a programming environment based on graphical objects rather than text listings. Graphics and photo images are viewed as objects to represent and present knowledge to minimize uncertainly and vagueness arose from the use of subjective and imprecise visual data. In summary, OOP environment makes it easier to develop a sophisticated knowledge-based advisory systems, which with a graphical user interface, is also easier to use.

Database Access

For most KBS users, data access is one of the most important requirements of KBS. A KBS with an interface connecting to a database by pointing and clicking is significantly more appealing to would-be users. Database connectivity should allow the users to perform operations such as open, edit and close on the database file. Information in relation to inspection and repair can be assessed and documented along with the degree of urgency and cost of maintenance actions. With the aid of database editing, the condition of the facility can be continuously monitored.

PAVE (Fig. 1) and TRUSS (Fig. 2) are being developed for the maintenance of pavements [Foo and Akhras 1993] and timber trusses [Akhras and Foo 1993] respectively. Both systems permit direct read/write operations on dBASE III files. "Information display" in Fig. 1, and similarly in Fig. 2, displays information extracted from an external database file, while "database functions" allow the users to point, click and execute read/write operations.

In both systems, database classes are automatically assigned the properties SINGLE and EXTERNAL. The SINGLE property limits the class to a single instance or value. EXTERNAL specifies the database product type (i.e., dBASE III) and filename (PAVE.DBF or TRUSS.DBF). The attributes and their attribute types correspond to the fields and field types in the external database.

User Interface

User interface plays a major role in the acceptance of any system by its end-users. A good system should have graphical user interface with explanatory facilities and little or no programming knowledge and experience should be required to conduct a diagnostic analysis.

In conventional rule-based KBS, user interface is rather linear and uni-directional. The user responds to a series of questions prompted by the computer, usually one at a time. After accepting the user's answer to a previous question, the computer analyses the situation before deciding which path or what the next question should be. User consultation is more or less controlled by the computer, and not by the user.

In many of these uni-directional systems and after answering a few questions, the user cannot go back to check the answers provided to the previous questions. The only way to do so is to restart the system. Although it may seem very minor, this limitation is very frustrating to any users, especially the experts. This situation may happen in more than one case. To answer in an appropriate way the present question, the user needs to know exactly what he already stated in the previous questions. The explanation for the present question provides more information on a particular aspect of the previous questions and the user would like to modify all or part of the input. Finally, the present questions may remind the user of new or different information to answer previous questions.

In short, uni-directional systems are not user friendly. On the other hand, a well designed knowledge-based advisory system provides the user with a dynamic user interface in which the user determines the direction or path that the computer should proceed to follow.

Both PAVE and TRUSS present to the users a list of possible problems (A in Fig. 1 and Fig. 2), which, by the way, is linked to the type of pavement or member under consideration. For example, there are a total of twenty-four (24) distress manifestations for concrete pavements or eight (8) possible defects that may apply to a horizontal bracing member. It would be quite tedious and confusing if the user were to be prompted a series of 24 questions, one at a time about each type of distress. The user would likely have forgotten whether "YES" or "NO" has been selected for question number (or distress type) 4 while question number 20 is on the screen.

Quite often, a dynamic link between multi-layered questions is needed, such as in PAVE where the type, severity and density of each distress are to be identified. The design of this link is important for both the acceptance and efficient use of the system by the users.

In Fig. 1, "C" links the type of distress to its density and severity classifications, whose selections are in turn linked to the "distress and treatment display". Density and severity classifications change with respect to the type of distress selected in "A" of Fig. 1. The dynamic link between the type of pavement (asphalt, concrete or surface-treated), the applicable list of distress manifestation and the explanatory facility are shown as "B" in Fig. 1. Shown in "A" of Fig. 1 is a list of 24 distress types for concrete pavements. Should the type of pavement be asphalt, "A" in Fig. 1 will be replaced by a list of 15 distress types and the explanatory facility or window would be for asphalt pavements as well.

Similarly, for TRUSS, there is a dynamic link between the member identification (HB or horizontal bracing in Fig 2), list of defects ("B" in Fig. 2) and the explanatory facility. If a different member such as a vertical member is under consideration, a list of nine defects applicable to vertical members would be displayed in "B" (Fig. 1) and "Information on Defects" would be for vertical members accordingly.

User help screens not only are important for the acceptance and efficient use of the system but also enhance the use of the system as a training tool. If the system user chooses to receive advice about the type or classification or repair of a problem or defect, information should be readily available and is displayed on how to interpreted the information. Some form of pictorial or graphical information should be supported in addition to text.

Use of graphical representation is easier and quicker to understand than lengthy documentation. As the users may be using the system in the field, conventional method of knowledge presentation, such as lengthy text explanations, should be avoid. As shown in Fig. 1, a comprehensive sets of photographic examples and detailed description of all types of distress manifestations as given in relevant guidelines [Chong et al 1989] can be displayed. Fig. 2 presents to the user graphical knowledge of not only what is and how to measure "grain slope" but also the repair details for a particular member [DND 1988].

Hypertext and hypermedia capability

The key feature of the typical information-handling problem encountered during an inspection is the correct interpretation of visual images. In order to lessen the dependency on subjective judgement and to improve the consistency in damage assessment and decision making, a diagnostic knowledge-based system with dynamic graphical representation is very useful.

Hypertext [Conklin 1987], an application consisting of a series of interconnected windows, displays, text, and graphics that permits the end user to pursue information in a non-linear, self-directed manner, provides graphic-oriented interface features such as interactive graphics and graphical representation of knowledge base by means of mapping of graphic displays to and from conclusions. With dynamic and graphical representation of knowledge capability, the systems should allow a much more precise representation of complex relationships and the knowledge could be learned directly from experience or examples.

The systems should also support the hypermedia paradigm or multi-media access in which perceptual objects and their relationships can be manipulated by directly manipulating windows and icons. Current technology does not limit the representation of these objects to static boxes on a CRT screen. Advances in animation, colour, 3D displays, sound etc will keep making the display more active and realistic, the data represented richer and more detailed, and the input more natural and direct.

Conclusions

Engineers, who are responsible for the well-being of our economic resources such as equipments, buildings and infrastructures, will be obligated to meet higher and new demands by improving the efficiency of their operations. As such, engineers have developed knowledge-based advisory systems and solutions corresponding to the needs of the profession. A knowledge-based advisory system is as much a problem-solver as an intelligent assistant and a training aid which is being regarded as a major application of knowledge-based systems in engineering diagnosis and operations in the future.

Fully integrated decision-support and training systems combine knowledge-base systems with interactive audio/visual systems and other conventional media. The knowledge and experience gained from developing two object-oriented knowledge-based systems confirms that knowledge-based systems have good potential for capturing knowledge and building decision-making/training systems for the diagnostic process.

Acknowledgements

The financial assistance of the Department of National Defence of Canada is gratefully acknowledged.

References

Akhras, G. and Foo, H.C., [1993] "Timber truss inspection using a graphical object-oriented system", *Journal of Microcomputers in Civil Engineering*, vol.8,no.5,pp.329-341.

Chong, G.J., Jewer, F.W. and Macey, K., [1989] "Pavement maintenance guidelines : distresses, maintenance alternatives, performance standards", Report SP-001, *Ontario Ministry of Transportation and Communications*, Ottawa, Ontario, August 1989.

Conklin, J., [1987] "Hypertext : an introduction and survey", *IEEE Computer*, vol. 2, no.9, pp.14-41.

Cox, B., [1987] "Object-oriented programming : an evolutionary approach", Addison-Wesley Publishing Company, Menlo Park, California.

Department of National Defence (DND) of Canada, [1988] "34 m warren type wood trusses and columns maintenance (C-98-006-002/TP-002)", National Defence Headquarters, Ottawa, Ontario.

Foo, H.C. and Akhras. G., [1993] "A graphical object-oriented system for the maintenance of concrete pavements", *Proceedings, International Conference on "Concrete 2000"*. Dundee, Scotland, September 7-9, 1993.

Goldberg, A. and Robson, D., [1985] "Smalltalk-80 : the language and its implementation", Addison-Wesley Publishing Company, Menlo Park, California.

Information Builders Inc., [1990] "Level5 object : user's guide", New York.

Pascoe, G.A., [1986] "Elements of object-oriented programming", *BYTE*, Mc-Graw Hill, Inc. New York, pp. 139-144.

Tesler, L., [1986] "Programming experiences", *BYTE*, Mc-Graw Hill, Inc. New York,. pp.195-210.

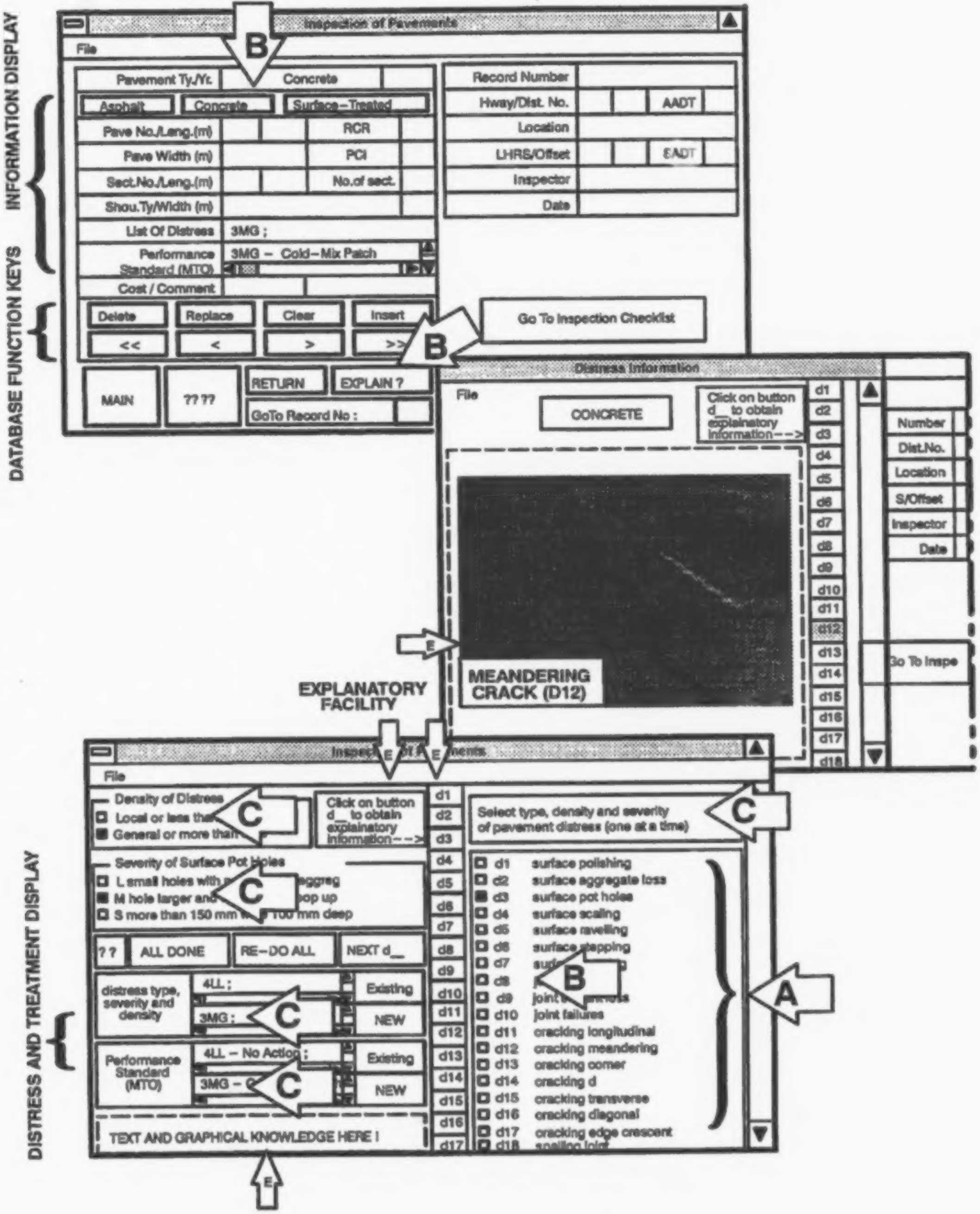
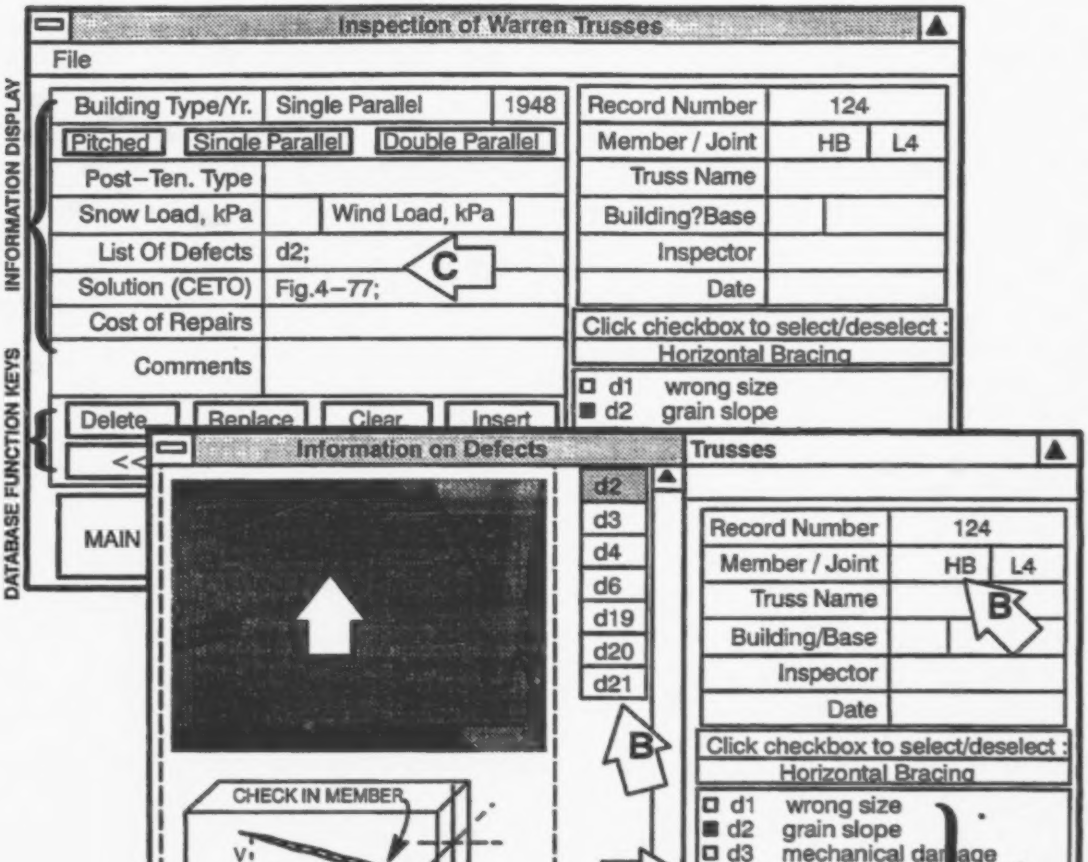

INFORMATION DISPLAY
DATABASE FUNCTION KEYS
Inspection of Pavements
File
B
Pavement Ty./Yr.
Concrete
Asphalt
Concrete
Surface-Treated
Pave No./Leng.(m)
RCR
Pave Width (m)
PCI
Sect.No./Leng.(m)
No.of sect.
Shou.Ty/Width (m)
List Of Distress
3MG ;
Performance Standard (MTO)
3MG - Cold-Mix Patch
Cost / Comment
Record Number
Hway/Dist. No.
AADT
Location
LHRS/Offset
SADT
Inspector
Date
Delete
Replace
Clear
Insert
<<
<
>
>>
Go To Inspection Checklist
MAIN
?? ??
RETURN
EXPLAIN ?
GoTo Record No :
B
Distress Information
File
CONCRETE
Click on button d__ to obtain explainatory information-->
d1
d2
d3
d4
d5
d6
d7
d8
d9
d10
d11
d12
d13
d14
d15
d16
d17
d18
Number
Dist.No.
Location
S/Offset
Inspector
Date
Go To Inspe
E
MEANDERING CRACK (D12)
EXPLANATORY FACILITY
E
E
DISTRESS AND TREATMENT DISPLAY
Inspection of Pavements
File
Density of Distress
☐ Local or less than
■ General or more than
C
Click on button d__ to obtain explainatory information-->
Select type, density and severity of pavement distress (one at a time)
C
Severity of Surface Pot Holes
☐ L small holes with aggreg
■ M hole larger and pop up
☐ S more than 150 mm 100 mm deep
C
☐ d1 surface polishing
☐ d2 surface aggregate loss
■ d3 surface pot holes
☐ d4 surface scaling
☐ d5 surface ravelling
☐ d6 surface stepping
☐ d7
☐ d8
☐ d9 joint
☐ d10 joint failures
☐ d11 cracking longitudinal
☐ d12 cracking meandering
☐ d13 cracking corner
☐ d14 cracking d
☐ d15 cracking transverse
☐ d16 cracking diagonal
☐ d17 cracking edge crescent
☐ d18 spalling joint
B
A
? ?
ALL DONE
RE-DO ALL
NEXT d__
distress type, severity and density
4LL ;
Existing
3MG ;
NEW
C
Performance Standard (MTO)
4LL - No Action ;
Existing
3MG -
NEW
C
TEXT AND GRAPHICAL KNOWLEDGE HERE !
E

Figure 1 Typical screen displays of PAVE



Inspection of Warren Trusses

Building Type/Yr.	Single Parallel	1948
Post–Ten. Type		
Snow Load, kPa		Wind Load, kPa
List Of Defects	d2;	
Solution (CETO)	Fig.4–77;	
Cost of Repairs		
Comments		

Record Number	124	
Member / Joint	HB	L4
Truss Name		
Building?Base		
Inspector		
Date		

Click checkbox to select/deselect :
Horizontal Bracing

☐ d1 wrong size
■ d2 grain slope

Information on Defects

CHECK IN MEMBER

SLOPE OF GRAIN = V/H

For grain slope greater than 1/10, remove and replace damaged member with new No.1 D.Fir of the same size. Drill and groove to match existing bolt and split ring locations.

Note : Provide shoring where necessary before removing damaged members.

Fig. 4–77

Trusses

Record Number	124	
Member / Joint	HB	L4
Truss Name		
Building/Base		
Inspector		
Date		

Click checkbox to select/deselect :
Horizontal Bracing

☐ d1 wrong size
■ d2 grain slope
☐ d3 mechanical damage
☐ d4 member decayed
☐ d6 end split to me
☐ d19 bracing long wide split
☐ d20 bracing short narrow split
☐ d21 bracing wedge condition

INFORMATION DISPLAY
DATABASE FUNCTION KEYS
File
Pitched
Single Parallel
Double Parallel
Delete
Replace
Clear
Insert
<<
MAIN
C
d2
d3
d4
d6
d19
d20
d21
B
B
B
A
V
H
decking
vertical
vertical sway bracing
horizontal strut bracing

Figure 2 Typical screen displays of TRUSS

903228

Hybrid Artificial Intelligence and Military Applications

James E. Bowen, Jianli Liu and B.A. Bowen
CompEngServ Ltd.
Suite 300, 19 Fairmont Ave,
Ottawa, Ontario, K1Y 1X4

#149687

Abstract

The paper reports on a project sponsored by the Defence Research Establishment Val Cartier which was devoted primarily to exploring the potential for combining neural networks and knowledge-based systems into a composite system which would extend or enlarge the capabilities of either paradigm to address the requirements of command and control applications. The objectives of this project were thus to undertake a study on the integration of neural networks to knowledge-based systems in order to develop better decision support systems and to evaluate the usefulness of a distributed knowledge-based system shell which integrates the neural network paradigm with a knowledge-based system.

This paper will begin by concentrating on the strengths and weaknesses of expert systems and neural networks. This will provide the bases for finding applications in which they could complement each other.

This paper demonstrates that neural networks and expert systems can be successfully combined to yield decision support systems which have capabilities in certain applications and which exceed those obtainable when either paradigm operates alone. In most cases, the combinations have been a single neural network augmented by the capabilities of an expert system or vice versa.

Finally, the paper will concentrate on deriving suitable and useful plans for research and development projects in this field for DND. The paper divides the R&D projects in to two major parts: fundamental problems, and applications.

The major problems of a theoretical nature revolve around the representation of the desired system in terms that permit a mapping onto the appropriate paradigms and the transfer of data and control .

At the more applied level many applications will be proposed, the field of Resource Allocation, Scheduling and Logistics seemed to be the most fruitful.

Introduction

Command and control systems are characterized by a global decision process, which in response to input data, arrives at a final decision. The process of decision making can be routine or complex. In routine situations, a few alternatives are evaluated depending on the input data; while in more complex situations, input data must be evaluated, compared to internal data, alternatives prioritized and an optimal response chosen. Traditionally, the decision processes, while potentially complex, were algorithmic in nature. Modern systems often include some form of artificial intelligence or machine learning paradigms to enhance the decisions or to enlarge the range of decisions to reflect expert human-like behavior.

In most command and control applications, the global decision process can be decomposed into a set of sub-decisions that cooperatively execute to arrive at a final course of action. In the general case, the system can utilize different sub-decision processes such as algorithms, neural networks, and/or expert systems depending on the particular task.

The current challenge is to extend the implementation possibilities to include the most appropriate paradigm for arriving at the required decision.

Neural Nets and Expert Systems

The purpose of this section is to characterize the differences between neural networks and expert systems in such a fashion that, given an appropriate characterization of a decision problem, an optimal assignment of responsibility can be allocated to either paradigm.

In general, the two approaches can be divided into the symbolic and connectionist.

The symbolic level manipulates concepts which are individually and explicitly defined by the human expert. The concepts can be simple, or very complex, and the reasoning steps are expressed in rules, semantic nets or frames which relate combinations of concepts (i.e., the antecedents of rules) to their consequences. The inferencing mechanism involves searching the rule base to establish antecedent-consequence linkages (forward chaining) or vice versa (backward chaining). Each rule could be considered as a functional unit, which forms part of the over-all knowledge contained in the rule base.

The connectionist approach, on the other hand, uses functional units which behave identically by computing a 'closeness' metric (e.g., an inner product) of the input vector and an internally stored vector (often called the weight vector) followed by a nonlinear weighting function (often a sigmoid). Individual units differ only in their weight vector. It is these weight vectors and the interconnection of the functional units that constitute the knowledge of the network. This knowledge is obtained or instantiated during the training phase by learning from examples. The entire processing activity of a connectionist network is numeric. The input vector must be a set of numbers and the data-flow along the arcs connecting the functional units is a number. The internal processing of each neuron is a calculation yielding a number.

The following section outlines major distinguishing characteristics of the two approaches focusing on knowledge acquisition and representation.

Expert systems tend to be domain specific and function extremely well when problems are well defined and rules are clearly articulated. They can be flexible in gathering responses and data.

In terms of knowledge acquisition, several difficulties can be encountered:

a. The expert may have difficulty expressing the rules in a form compatible with the representation mechanism;

b. The expert may have difficulty creating the total range of responses that are necessary, until the situations actually occur. Many iterations and intermediate prototypes may be necessary to find a complete picture;

c. There is often a hierarchy of abstractions of situations in the expert's domain to which they evaluate and respond intuitively. The components and structure of this hierarchy is often difficult to determine;

d. If several experts are involved, they may have different views of how they respond to a situation, which may seem contradictory but are nevertheless effective. These views or approaches are often difficult to understand and to merge;

e. Merging two sources of knowledge can often be difficult depending on how the experts have constructed their heuristics to either augment the formal rules or to find mechanisms to accomplish their goals by circumventing them; and,

f. The expert may respond to many situations in a manner which is highly intuitive, and very difficult to represent by any known mechanisms.

Knowledge acquisition directly from experts can, therefore, be an iterative, lengthy and expensive process depending on the availability and source of the knowledge, the inferencing process, the size of the domain, and range of cases that must be reckoned with.

In terms of knowledge representation, the knowledge of an expert system is represented by rules composed of antecedents and consequences, and in some cases a certainty factor. Normally, the input data provides the beginning point of a trace through the rules linking a consequences to antecedents (backward chaining) or vice-versa (forward chaining). The possible trace paths through the rules can be very simple or very complex depending on their number and on the complexity of the decision process. The position of the rules in the knowledge base can affect performance (because of the search time required to find the rule) and in some cases affect results when the order is changed.

Successful applications have the following general characteristics:

a. An algorithm to solve the problem is unknown or expensive to discover;

b. A written set of rules and regulations are available;

c. An expert is available who can articulate the rules and heuristics used to solve the problem; and,

d. The problem domain is well focused and reasonably narrow.

Neural networks have a broad response capability based on their capacity to provide a classification of a set of inputs, even though a precise description of the problem is not available, and when the input data is low quality, unreliable, redundant, time decaying and incomplete.

Neural networks can capture knowledge easily by generalizing the given training date and capture a large number of cases quickly to provide an acceptably accurate response. This requires many representative examples for successful training.

In a trained neural network, the knowledge of the network lies in the weights of each neuron and in the topology of the interconnections. This knowledge is learned and stored during the training sessions. An obvious problem is that this knowledge is in the form of connection weights and topology, and it is very difficult to interpret.

Successful neural network applications tend to have the following general characteristics:

a. An algorithm to solve the problem is unknown or expensive to discover;

b. Heuristics or rules to solve the problem are unknown or perhaps difficult to enunciate; and,

c. The application is data-intensive and a variety of data sets are available, which describe specific examples, and which can be identified as correct.

The above discussion bypasses issues associated with the user interface, design and implementation, validation, production standards and maintenance issues.

Hybrid Architectures

In general, the requirement for hybrid architectures can be based on the requirements:

a. A range of different expertise (i.e., a group of experts) may be required to solve the problem;

b. The decision processes may derive their results by any means thought appropriate, including artificial intelligence, connectionist, algorithms, or a sensor assessment of the outside world;

c. Independent parallel decisions may have to be integrated, assessed and weighed to form an intermediate decision for further consideration by on-going processes;

d. Iterations may be necessary as the process proceeds, due to newly acquired data or as a result of the process itself;

e. The decisions may be temporally dependent; and,

f. The input data may have an associated uncertainty and/or reliability, and the effects of this uncertainty must be propagated to the final output.

The direct integration of neural networks and expert systems is not straight-forward for a variety of reasons:

a. The nature of the data in a neural network is numeric, while in an expert system it is symbolic;

b. Relations in neural networks are causal and represent associations, while in expert systems relations are structured and allow combinations; and,

c. The characteristic behavior in a neural network is self-organization, while in expert systems the combinatorial capability of structures is the major feature.

There are two basic integration approaches: Autonomous Functional Units and Embedded Sub-Systems.

Several different systems for the classification of hybrid architectures of neural networks and expert system have been proposed. Among them, the system proposed by [Caudill, 1991] and the system proposed by [Medsker & Bailey, 1992], are typical and represent different views of the classification of the hybrid architectures.

In a complete classification system, the functionalities of expert systems and neural networks in the hybrid architectures and the characteristics of software architecture should all be considered. To make up the deficiencies of the above classifications, we proposed a new classification system. The proposed system includes seven classes: autonomous functional units; closed embedded sub-systems; open embedded sub-systems; simulating externally; simulating internally; transformational systems; and, developmental tools. In defining these classes, we considered the difference in the functionality of expert systems and neural networks in the hybrid architectures.

Autonomous Functional Units

This class is the mainstream in the study of the hybrid architectures of expert systems and neural networks. In this class, each expert system or neural network is an autonomous functional unit. In a decision procedure, each decision node is implemented by an expert system or a neural network independently. This class is applicable for the cases where a large and complex problem can be broken into small sub-problems; each sub-problem would be solved by individual expert systems or neural networks.

In implementing the hybrid architectures of this class, the data coupling between an expert system and a neural network is important, since it affects the efficiency and performance of the final system. There are basically two coupling approaches: loose coupling; and, tight coupling.

Loose Coupling

The loose coupling approach is characterized by a data sharing mechanism involving external data structures. The storage of intermediate data and its formatting for use by the next process is done in the data structure.

Tight Coupling

In this architecture, the expert system and neural network share common memory resident data (instead of passing data structures or using external data files). This improves the interactive capabilities (compared to the loose coupling approach) and enhances performance. Tightly coupled systems often appear to have the same architecture as a loosely coupled system, however, if properly designed, they are faster because data is shared directly (without intermediate external data structures or transformations).

Embedded Sub-Systems

In this class, one system is embedded within the other to implement some of the functions. Of the latter, however, the embedded system is transparent to the user.

There are two known, typical approaches in this class:

Neural Networks as the Rule Firing Mechanism in Expert Systems

The basic idea is to bury a neural network inside the inference engine of a rule-based system and use the neural network to perform highly efficient and intelligent selection of the next rule to fire. The obvious benefit would be that the resulting expert system would run faster and select more intelligently the correct rule to fire in cases of multiple matches.

The approach would benefit very large rule bases in which sequential rule-matching can take a long time, and in systems that frequently have several rules that match the current state. The rule-based system would leverage-off the parallel nature of the neural network to improve its efficiency and performance.

Neural Networks as the Knowledge Base in Expert Systems

One of strengths of neural networks is their associative memory capability, which can be utilized to build the knowledge base in an expert system. The neural network memorizes the facts during its training. Within the expert system, the neural network's input and output nodes are used as the outlets of the facts. Each instance of the pair of input and output patterns provides a fact to the expert system.

The embedded neural knowledge-base efficiently and compactly stores a large volume of facts. However, the issue of knowledge-base updating in such an integration architecture can be difficult, depending on what type of neural network is used. If the incremental retraining (accepting new facts, with no need of retraining all previously accepted facts) is possible, the incremental updating ability of the neural knowledge-base clearly is an advantage of the integration architecture. If the incremental retraining is not allowed (e.g., as in the back propagation neural net), this becomes an obvious drawback.

Open Embedded Sub-Systems

In this class, one system is used as the main system, the other used as the secondary system to be called when needed to improve the performance of and/or to supplement the capabilities of the main system. Both the main system and the secondary system are invisible to the user. This class differs from the autonomous functional units class in that the hybrid architecture of this class only implements one node in a decision procedure.

Typical hybrid architectures in this class include:

Expert Systems Interprets The Results of Neural Networks

In the complete system, an input is presented to the network and the network provides the correct response during normal operation. When the user asks for an explanation, the expert system considers the original input pattern as well as the neural network's answer. It then backward chains from the answer to the input pattern, constructing a plausible chain of reasoning to explain why this answer may be correct. In essence, it rationalizes an explanation on the spot, making it correspond with the neural network's response.

This sort of rationalization response provides the user with reassurance that the neural network is operating properly without relying on the rule-based system to generate the solution. The rule-based system does not have to be detailed or complete; it only has to be complete enough to justify the answer to the user. This sort of post-response rationalization is probably a lot closer to the way people solve problems than we would like to admit.

Simulating Externally

In this class, a hybrid architecture appears to be a network. Internally, each node of the network is a symbolic information processing mechanism, which can be as simple as a concept node in the knowledge structure built in an expert system, or as complex as a complete expert system. Externally, the network is a mimic of a neural network, two nodes communicates through the propagation of numeric information along the interconnection between them, for example, connectionist expert system systems [Gallant, 1988].

Connectionist Expert Systems

Connectionist expert systems, in general, rely on local knowledge representations, as opposed to the distributed representation of most neural networks, and they reason through spreading activation. Connectionist expert systems represent relationships between pieces of knowledge with weighted links between symbolic nodes. Such systems have been used in medical diagnosis, information retrieval and analysis, and pattern classification.

Simulating Internally

In contrast to the simulating externally class the representation and global inferencing mechanisms are simulated, the simulating internally class is concerned with the neural network's internal representation.

Basically, the way to find out the internal representation of a neural net is to first train the neural network to solve the problem and then reverse-engineer it to determine the features the neural network is using to make decisions. For example, in the training of a neural network for an alphabet-recognition task, we could force some of the neurons in the middle layer to look for obvious features like vertical straight lines or curves of various sizes and locations. These neurons then become the designated feature detectors. By analyzing the behavior of these feature detectors, it may be possible to determine how the neural network makes decisions.

Transformational Systems

The expert system is transformed into a neural network or vice versa. A typical situation might be that the development system was chosen as a neural network because the application was data-intensive and there was a need to generalize and filter errors in the data. The delivery of an expert system was chosen because of the need to document and verify the knowledge used to make the decisions, and justification of the decision was needed. On the other hand, an expert system could be converted to a neural network because it was incapable of solving the whole problem, or the speed, adaptability, and robustness of the neural network is required. Knowledge from the expert system is used to set the initial conditions and to derive the training set for the neural network.

Developmental Tools

In this class, one system is used to assist the development of the other. There are basically two approaches:

Neural Networks Extracts Expert System Rules

First, a neural network is trained to solve the problem at hand. Then, the trained neural network is used to generate a set of rules based on the given input data. Finally, the generated rule-set is used to construct the rule-base of an expert system.

In effect, the neural network is trained to be the expert in the expert systems development. The obvious use of this procedure is for problems where no expert exists to construct a rule-based system, but where the properties of a rule-based solution are desired.

Expert System Controls Neural Network Training

There is no systematic theories for the conduct of neural network training. In practice, training is based on the trial-and-error method. Thus, an expert system is constructed with the knowledge of controlling the neural network training built-in. In the development of a neural network, the expert system is used to intelligently control the training of the network in terms of offering recommendations for the appropriate selection of neural network training parameters, the number of layers, the number of neurons within a layer, etc., according to the given problem.

Examples of Hybrid Architectures

The literature contains many examples of hybrid distributed architectures.

A Resource Projection System

Hanson and Brekke [1988] developed a system for projecting manpower and resource requirements for maintaining networks of workstations at NASA. The overall function of the system was to estimate the resource requirements and the completion time for the services requested by customers.

The major part of the required output is the allocation of resources, which can be obtained using an expert system. An expert system can properly allocate the manpower and resource for the service requested, based on the knowledge (rules) acquired from managers and technicians. However, another part of the output is an estimation of the completion time for the activities required, which will be more easily obtained by using a neural network. A neural network can be trained using historical data, in which the time needed for different types of activities in the past services are present. Thus, the complete system output can be obtained using an expert system plus a called-up neural network.

A Process Control System

A process control system [Hendler, 1991] was designed for the control of mechanical systems in which sensor feedback is needed to monitor the status of ongoing processes. The overall function of the system was to track changes in the sensor signal to detect error conditions in system's operation so corrective actions can be taken before serious errors are committed.

The system uses a neural net to accomplish the error condition classification task.

The system uses an expert system to accomplish the decision making task. Making intelligent decisions based on the received sensor signals is very different from classifying these signals. Such decisions may require merging a long time history of sensor results; all features of a situation may need to be present before expensive corrective operations would be authorized; the set of all possible problems may not be known, the recognition of a problem is often context dependent; and, hierarchical structure is often used in the decision making processes.

Military Research and Development Projects

In the following section the projects are divided into research issues to be addressed and projects which could have immediate applications.

Design Related Issues

Most professionals who solve problems of any complexity rely on a global inferencing strategy for working out their solutions. This strategy can be very complex, and rely on a hierarchical decomposition of the problem, which in turn can be iterative and involve a variety of mechanisms for resolving the issues to be considered at each step.

Some questions related to design issues suitable for further research are outlined in the following six sub-sections.

Categorization of Applications

What type of applications require the use of hybrid combinations of artificial intelligent paradigms?

Such applicability should be determined on the case-by-case base in practice. And a mechanism for doing so is needed.

Decomposition of Problem

How is a problem partitioned into a number of sub problems such that each sub problem is allocated to one of the paradigms?

While, traditional AI approaches (expert systems) are suited for knowledge-based tasks, while neural networks seem to be appropriate for behavior-based tasks. However, a complete set of down-to-earth guidelines needs to be derived.

Global Inferencing

How is control responsibility determined and assigned to the components in the system? Various options are available such as central control, and, distributed control (cooperative inferencing).

Knowledge Representation

How is knowledge represented in the system?

While the knowledge representation in a neural net is numeric, in an expert system it is symbolic. Research could examine hybrid systems consisting of case-based reasoning, neural nets and expert systems,

Incremental Learning

How does a paradigm (such as a neural net) in the hybrid system adapt to the changing environment?

In a changing environment, the retraining (incremental learning) of the neural net is often needed. How to manage this retraining? Will it be done on-line or off-line? There exist a variety of options (in fact, none of the existing options represents a perfect solution), depending on the requirement to the system.

Data Exchange and Control Between Paradigms

Data transfers between paradigms generally require that data be coded (from symbolic to numeric) and decoded (numeric to symbolic). This is not an easy task since for example, and an expert system querying a neural network for a decision of the appropriate partition of a data vector must code the information so as to preserve the closeness criteria used by the neural network.

The derivation of any means of preserving the information content of the vectors as they are passed between paradigms will be a contribution.

Human Machine Interface (HMI) Mechanisms

There are several research areas possible in HMI as outlined in the next three sub-sections.

User Interface

How does a user interface represent the problem domain externally (note that this representation issue is different from the knowledge representation issue)?

The requirements for the use of symbolic representation are:

a. Symbols must be interactive. As output entities, they depict the problem domain in its current state. As input entities, they are handlers, knobs, and other gismos whereby the user manipulates the problem domain.

b. The symbols are not isolated entities. There exist the spatial and temporal relations among the symbols, and the dynamics of the symbols, which must be characterized in the representation.

Interaction Control

The interaction control issue in modeling a user interface is concerned with deciding the constructs of the interface.

The examples of the choice constructs include menu-driven systems and user-driven systems.

Multiple Entry Points

Hybrid artificial intelligent systems involve paradigms which execute independently. To be able to effectively and efficiently consult different paradigms in the system, the following properties of the user interface are expected:

a. Commonalty across Paradigms: Similar functions (interface services) are implemented similarly from one paradigm to other.

b. Freedom of Navigation: The user can freely gain and release the control to any paradigm.

System Interfaces

The hybrid system must, as an entity, interact with other software components of the system (e.g., data bases, operating systems, etc.) and potentially with distributed elements of the larger system. The issue here is to maintain the proper logical control constrained by the interface requirements of the existing platforms.

Resource Allocation, Scheduling and Logistics

This is a board research area where many potential applications exist. It is an area also in which the payoff is easy to demonstrate. Taken from a sufficiently broad point of view, this topic covers all the others mentioned in this report. Applications range from assigning vehicles in a car pool to weapon-to-target allocation search and rescue requirements, scheduling and planning of convoys, and planes-to-missions scheduling.

This capability is also a powerful teaching and/or training aid.

Impact Analysis

This application would serve in a module to augment a resource allocation or a planning system. By combining AI paradigms with algorithmic modules the impacts of perturbation of an on-going plan can be determined. Impact analysis essentially answers the question "what if?". This is a much more complicated problem than computing a cost-function and choosing the optimal solution. The system must be designed to permit the exploration of what-if scenarios, and the HMI must permit their input. One possible usage would be assessing tactical air capability during various situations.

Threat Evaluation and Ranking

This function is a subset of a broader field of situational analysis. The goal is to assess a view of a situation and determine its significance. This application could monitor incoming attacks, e.g. missiles, vehicles, etc. and determine its threat potential and assign rankings for a targeting or response system.

Emergency Response

Emergency response plans can be formulated and a hybrid system can disseminate recommendations for an appropriate response for situations such as fire, floods, insurrections, etc. Often these calls demand very short response times, and there is a lack of experience in formulating a quick response. The maintenance of the corporate memory from occasion to occasion would be useful.

Correlation

Such a system could monitor flight plans with track and assist in ensuring aircraft in Canada's airspace are behaving as expected. This would be useful in drug interdiction.

Electronic War Games

Such a system would assist in develop various strategic and tactical plans. By incorporating "knowledgeable opponent responses" various aspects of plans could be developed and tested. In addition, command expertise could be developed in junior officers.

Filtering of Intelligence Data

Such a system could monitor various data sources and detect significant information and provide analysis.

Air Traffic Control

An air traffic control advisor could be constructed to assist in controlling an airspace filled with planes. The system could predict airspace conflicts and recommend avoidance maneuvers.

Multiple Sensor Data Fusion

As more and more sensors are utilized of varying functionality, the need arises to automatically monitor incoming data and alert operators to potential concerns and give potential courses of action. Applications extend from the cockpit of a jet to the control room of ship. The need is to transform the input from arrays of sensors into useful information which will aid in formulating the appropriate response.

Weather Prediction

By combining the expertise of weather personnel with the pattern recognition abilities of neural nets, automatic weather prediction and interpretation might be possible.

Detection and Identification of Targets

A great deal of work has been completed in this area mostly on the basis of "pattern recognition" however, a system could enhance such pattern recognition approach by the augmenting it with knowledge or case-based intelligence.

Resolution of Frequency Jamming

With the high need for communications and electronic equipment, add on modules which attempt to diagnose and correct frequency problems would be beneficial.

An Integrated Prototyping and Test Facility

An essential requirement to the assessment of new paradigms, platforms and concepts is a flexible hardware/software platform for simulation and proof-of-concepts.

For example, HMI is a source of constant concern to system designers. Capabilities for implementing the information interchange between humans and machines extend beyond graphic screen displays using various windows to more exotic devices such as pens, touch screens, and even voice.

Training

In this context we are referring to training in the particular situational response, (not to training on the use of a system).

Developing appropriate responses in a complex environment becomes an ever increasing cost in almost any organization. In an automated decision support system, a worthwhile design goal is the inclusion of modules that permit experienced people to set up scenarios, which are solved 'off-line" by neophytes.

Conclusions

This paper demonstrates that more complex decision problems can be implemented by combining the artificial intelligence paradigms. The paper demonstrates possible approaches and successful applications. It is apparent that a great many decision problems can be tackled in the military environment with hybrid artificial intelligence.

Acknowledgment

We wish to acknowledge and thank the Technical Authority Mr. Serge Roy and Mr. Donald Smith of DREV for discussions of Command and Control problems which helped focus our efforts throughout the project.

References

Caudill, M., [1990] "Expert Networks," *Byte*, pp. 108-116, October, 1991.

Gallant, S. I., [1988] "Connectionist expert systems," *Communications of the ACM*, vol. 31, pp. 152-169, 1988.

Hanson, M. A., and Brekke, R. L., [1988] "Workload Management Expert System - Combining Neural Networks and Rule-Based Programming In an Operational Application," *Proc. Instrument Society of America*, vol. 24, pp. 1721-1726, 1988.

Hendler, J. and Dickens, L., [1991] "Integrating Neural Network And Expert Reasoning: An Example," *Proc. AISB Conf. on Developments of Biological Standardization*, 1991.

Medsker, L. R., and Bailey, D. L., [1992] "Models and Guidelines for Integrating Expert Systems and Neural Networks," *Hybrid Architectures for Intelligent Systems*, Chapter 8, pp. 155-171, 1992.

905878

Development of a Vision Guided Robotic Assembly Workcell

Roy Ball & Terry Folinsbee
IMAGO Machine Vision Research Inc.
1354 Wellington Street, Ottawa, Ontario, K1Y 3C3

Tim Pryor
Sensor Adaptive Machines Inc. (SAMI)
6360 Hawthorne Drive, Windsor, Ontario, N8T 1J9

#149688

Abstract

IMAGO and SAMI are building a proof of concept workcell in order to demonstrate the feasibility of using machine vision for positional control of sheet metal components in the assembly of car bodies. This project is being performed under contract to the Canadian Space Agency through the Strategic Technologies in Automation and Robotics (STEAR) Program. The technology has the potential to be incorporated into the evolutionary design of the Mobile Servicing System (MSS), Canada's contribution to the international Space Station Freedom Program.

The manufacture of most car bodies today involves robot welding of smaller stampings into components. The process uses hard fixtures which are rigid and accurately surveyed posts or arches to which the stampings are clamped to define their positions in space. Blind robots are then programmed to go to a coordinate in space and spot weld two stampings. The components are then moved to the next hard fixture and so on until the assembly is completed. In this system, each hard fixture can only handle one basic body design. Thus changing the body design is costly and time consuming.

The proof of concept workcell being developed by IMAGO and SAMI will demonstrate how car bodies can be constructed using fixtureless assemblies. Specifically, it will demonstrate that car body parts can be brought into register with a very high degree of accuracy.

IMAGO is contributing its natural feature detection and tracking technology, the National Research Council is contributing its photogrammetry technology and SAMI is providing structured light technology and performing the system integration.

The Autobody Assembly Problem

The body of most production cars is an assembly of several major components such as the floor pan, quarter panels, and roof. Each of these major components is, in fact, a sub-assembly consisting of smaller sheet metal parts.

With today's technology, these assemblies are typically manufactured using 'hard fixturing'. The components to be assembled are placed loosely on a heavy metal fixture either by hand or more commonly by a robot. Clamps are then engaged to hold the pieces firmly against specific locating blocks to correct for misposition relative to each other.

These components typically have 'weld flanges' which, when overlaid on top of each other, are spot welded together to form a single assembly. In the welding operation, the weld gun provides two functions. The first is to apply pressure to both flange surfaces which ensures that they are firmly in contact with each other, and the second to actually fuse the two panels together.

This method of 'hard fixturing' has several limitations and weaknesses. Perhaps the most serious drawback is that most fixtures are designed to produce only one unique assembly, or assemblies having only subtle differences. More common than not is the scenario whereby each assembly requires its own unique fixture.

One might expect that these fixtures, with their typically massive structures, would generate near perfect assemblies each time. However, in practice this is not always the case. Part locators wear through use and, even occasionally, get displaced from their optimum position. Individual sheet metal parts can be misformed and robots can misposition parts.

Robots, generally, follow preset paths in loading and unloading of parts. They are typically unable to either detect or compensate for fixture or part errors. These problems are serious enough that the automotive industry has invested heavily in 'check fixtures' which are used to verify dimensional accuracy of separate subassemblies and/or major assemblies. This checking equipment is both expensive and in most cases, part specific. Therefore, it can only be justified for high volume or critical components.

The robots themselves also add some degree of inaccuracies to the operation. Robots vary in positional accuracy capabilities depending on the type of robot used, as well as from which manufacturer it was purchased. In addition, the 'part holding' tooling on their end-effector for pick up and holding of parts while being transported adds yet another variable.

Therefore precision part locators and clamps within the fixture are required to accurately locate and hold parts while being welded. The positional inaccuracies not only apply to the robot which is used to transport the part from place to place, but also the robots performing the actual welding of the components. Currently, 'weld' flanges on sheet metal parts, are made extra wide to accommodate this lack of accuracy. This requirement not only adds cost, but significant extra weight to the final product.

The objective of SAMI and IMAGO is to improve the current robot and part positioning capabilities, to the extent that hard fixturing could potentially be totally eliminated. A vision system will be used to determine the actual positions of mating sheet metal parts and, optionally, the positions of the robots themselves. From this information, error signals can be calculated and applied to the robot controllers so that parts are moved more accurately into position relative to each other. In addition, enhanced positional guidance might also be provided to the welding robot, thereby allowing for weld flanges to be reduced in size.

In addition to the above mentioned enhancements, fixtures could become much more flexible making them capable of handling many part assemblies. This would then reduce the large number of unique fixtures currently required and eliminate many of the associated costs. Assemblies could be inter-mixed on the assembly line allowing many low volume car body styles to be produced on a single production line.

An added benefit is that the verification of part position just prior to welding will lessen the need for checking stations after welding. This verification applies to all parts, not just the critical and high volume ones for which specialized check fixtures are now purchased.

Project Objectives

The global objective of this project is to demonstrate the feasibility of using machine vision for positional control of sheet metal components in the assembly of car bodies.

At the onset of the project, three specific technologies were considered to be primary in achieving this objective. NRC's photogrammetry, [5] SAMI's structured lighting, and IMAGO's feature finding [1]. With the overall objective in mind more specific project objectives are defined as follows:

1. Locate Parts Using Natural Features

   Ideally the part location would be determined by locating only natural features on the part. Many parts, particularly interior parts have an abundance of holes and structured (shaped) surfaces which can be used.

   As a goal, natural features will be grouped into types and appropriate algorithms designed to extract their positions. As far as practical these algorithms will remain generic in design so as to cover as broad a spectrum of features as possible.

2. Locate Parts Using Structured Light

   With structured lighting, the surface of a part is illuminated with an artificial pattern or target. Most commonly, the light source is a laser, and the pattern is a sheet of light forming a straight line on the part surface. When viewed at an angle, the straight line is seen as the contour of the part. The line of light is analyzed to determine range information.

   Structured light is essentially an extension of stereo imaging. One of the stereo cameras is replaced with a light projector and the reflection from the surface of interest is imaged in the remaining camera. The mathematics for calculating the position of the point on the surface that is illuminated by the light source is the same as if a stereo pair of cameras were used instead of the light projector/camera combination.

Our intent is to employ this technique to project artificial targets on the part. These have the advantage of not increasing the automotive manufacturers part process requirements and are non-marring.

There is a significant difference between the structured light targets and the natural targets which must be addressed in this project. Natural targets are fixed to the part so there is a direct relationship between target position and part position. Structured light targets are not fixed to the part.

3. Find Features In A Volume of Uncertainty

The parts used in the project are normal production parts as supplied by an automotive manufacturer. These parts are shipped and stored in carriers or at least stacked in an organized manner so as not to be damaged. They are not scattered randomly in a bin.

For that reason the initial part location is assumed to be within 50 mm and six degrees of nominal. Thus, search windows need to span only a volume of these dimensions when initially finding the part.

Once the part position is determined, it can be picked up and moved by a robot from one location to another. Its position once moved can be predicted with an accuracy subject only to the inherent robot positioning errors. Typical robot errors are much below 50 mm, thus 50 mm and six degrees of nominal define a working volume for this project.

4. Maintain High Accuracy

The automotive manufacturer designates reference locations which are distinct features such as large holes on an underbody to define the world (global) coordinate system for the body. One of these features will be the origin of the vision system. In that frame of reference, commonly quoted tolerances for part features are +/- 2.0 mm. In some cases, tolerances are specified as low as +/- 1.0 mm.

Each sensor (pair of cameras) works within a 'local coordinate system'. The information gathered by them has to be transformed back to the world system whenever results are shared between sensors. Thus, the tolerance limits for the local systems must necessarily be higher in order to allow for additional error during the transform.

Given the above, to meet overall accuracy requirements on the most critical features, the system should achieve accuracies within the world coordinate system of +/- 0.5 mm, and local accuracy +/- 0.2 mm. Resolution will be 0.1 mm.

5. Interactively Train The System

When a new assembly task is to be developed it should be possible to train the system to recognize actual parts interactively. The system should be able to learn to recognize and determine the location of these parts without adding new code or requiring input from a machine vision expert. As well, the system should be easy to operate.

IMAGO Testbed

For their part of the project, IMAGO used their feature matching software testbed. This uses similar algorithms to those used in the IMAGO XXX series of video target trackers. The IMAGO 100 is a rule based video target tracker which has been successfully used at a number of military trials [1,2,4]. The IMAGO 300 is a perimeter protection/electronic sentry version of the tracker which was evaluated recently at a Canadian Penitentiary [3]. One major difference is that the target trackers are implemented using special purpose hardware to achieve real-time performance. For the autobody project the algorithms are implemented entirely on a PC and operate an order of magnitude more slowly as a result.

This simplifies future transfer to the Space Station because there is no requirement to Space qualify any special purpose hardware and the processing speeds on the PC are adequate to accomplish many tasks in a Space environment where movements are much slower than in Terrestrial environments.

The IMAGO testbed includes the following capabilities:

1. Calibration. This software corrects for imperfect optics and to measure the transformations between the numerous coordinate systems involved.

2. Interactive Training. This allows the user to interactively build a computer model of a car part which consists of an image of each feature of interest and a three dimensional model of the location of that feature in a part frame of reference.
3. Location of Part. Each feature described in the part model is found using fast pseudo-correlation techniques. NRC's Space Vision System model based algorithms are used to geometrically match the found features to the 3-D model of the part to estimate its position and orientation [5].
4. Tracking. This is similar to location except the search volume is smaller and the processing speed faster.
5. Stereo photogrammetry for improved accuracy for final alignment. This was not fully integrated or evaluated since structured light is being used instead.

IMAGO Testbed Results

The testbed was completed in June 1992. Measurements indicated that the testbed was sufficiently accurate and fast to meet project objectives for Part Location and Tracking but that other technology would be required to obtain the very high final alignment accuracies for car bodies although the alignment accuracy would likely be sufficient for a number of Space Station scenarios.

The results are:

1. Autobody parts are automatically located in 5-10 seconds.
2. Subsequent movement of the parts is automatically tracked at about 1 Hz update rate.
3. The system has not yet been properly calibrated. Our results to date indicate that a calibrated system will have location accuracies of:
   (a) 5 mm in X and Y
   (b) 10-30 mm in Z (Range) depending on the size of the part
   (c) 1 to 2 degrees

We could significantly improve on the accuracy results by using interpolation to yield subpixel feature location and by using stereo capability of the testbed. This is unnecessary for autobody assembly since structured light techniques are being used to provide a more precise refinement of the IMAGO location capability.

The SAMI Structured Light System

The structured light portion of the system provided by SAMI is based on successful line projection based structured light systems sold for car body gauging worldwide during the period 1983 to date by SAMI's sister company, Diffracto Ltd.

Those particular systems project a fanned line of light about 40mm long at a given section on the body surface of interest, and utilize the distortion of the image of the line to tell the location of a given feature of the car body, such as a break line in the sheet metal, a lip around a door, or the like. Typically such corners, which comprise rolled edges, constitute the principle mating surfaces for which the outer panels of the body are attached, such as fenders, doors, hoods, etc., and these are often more critical than features of more clear cut shape and contrast, such as holes.

Because the present system is designed to be flexible to a large number of bodies and sub-assemblies thereof, a fixed line projection camera is not adequate, since the location and direction of light line projection is a function of the part to be assembled. For this reason a programmable light projector has been used to project a sequence of spots, which when taken together yield the same sort of result as the line projector, but without the requirement to illuminate a whole line. Certain key points on the intersecting surfaces of the body are typically sufficient to allow the body surface locations to be determined, and the orientation/location of the parts to be found. This can be done entirely from surface points without use of any natural target datums, or can be used more typically in combination with such datums, such as holes, special target points and the like.

The key is to do this accurately and quickly, so as not to consume a great deal of time making the measurements, eating into the cycle timing budget for the operation.

Present Integration Activities

At this point in time, the vision system, including a portion supplied from NRC and IMAGO and the SAMI structured light system are all being integrated together with a 4 axis Robot Positioner and a 6 axis welding robot (courtesy of Magna International), to provide co-ordinated vision guided part positioning.

This system will be utilized to demonstrate the ability to programmably mate two completely arbitrary sheet metal parts in correct juxtaposition relative to each other to the same or better tolerances as typically achieved by conventional hard tools.

Conclusions

Results from the integrated system are not yet available.

The IMAGO subsystem performance and limitations are:

1. Implemented on a PC with no special hardware.
2. Speed is suitable for slow operations which are typically found in space applications.
3. Terrestrial applications such as car body assembly require speed up using faster hardware similar to that used in the IMAGO XXX video target trackers.
4. Includes "part database" interactive training and photogrammetric calibration software.
5. Accuracy and reliability are sufficient to find and track the car parts.
6. Accuracy can be improved sufficiently to meet the requirements of many space and terrestrial applications. The autobody assembly requirements, however, dictate the use of structured light for final alignment.

References

1. Roy Ball, Lionel Pelkowitz, Steve Penzes Development of a Vision Guided Target Detector and Tracker, DND Workshop on Advanced Technologies in Knowledge Based Systems and Robotics, Nov. 1993.
2. Roy Ball, Terry Folinsbee, Lionel Pelkowitz, Activities in Machine Vision at IMAGO Manufacturing Ltd., 3rd Conference on Military Robotic Applications, Sep. 1991.
3. Terry Folinsbee, et al., Fast Response CCTV Image Recall and Motion Detection, 1993 International Carnahan Conference on Security Technology.
4. S. Penzes, R. Ball, L. Pelkowitz, Development of a UAV Video Autotracker, Proceedings of the 36th Annual Meeting of TTCP Technical Panel HTP-1 (Aerial Targets), July 1993.
5. Hughes, Pinkney, Basso, Perratt, NAE. Real-Time Photogrammetry Algorithm Development, Distribution is controlled by National Research Council of Canada.

90C229

DÉVELOPPEMENT D'UN CAPTEUR D'EFFORTS INTÉGRÉ POUR LE CONTRÔLE ADAPTATIF DE TRAJECTOIRES ROBOTISÉES

Alain Desrochers, Nicholas Krouglicof & Daniel Croteau
Département de génie de la production automatisée
Université du Québec - École de technologie supérieure
4750, avenue Henri-Julien, Montréal (Québec) H2T 2C8 CANADA
Téléphone : (514) 289-8800 télécopieur : (514) 289-9042
E-mail : alain@gpa.etsmtl.ca

#149C89

Résumé

Beaucoup de tâches industrielles usuelles requièrent un contrôle d'effort et une certaine souplesse de la part de l'ouvrier. On peut ainsi songer aux tâches d'assemblage, d'ébavurage, de ponçage, etc. Pour faire exécuter ces tâches par un robot, il est alors nécessaire d'équiper ceux-ci de capteurs d'efforts.

Dans cette perspective, nous avons développé un capteur d'efforts à six degrés de liberté basé sur une géométrie dite «plate-forme de Stewart».

Ce capteur d'efforts intégré trouve directement plusieurs applications. Ainsi, installé entre le poignet du robot et son effecteur, il permet, grâce à un programme approprié, de déplacer le robot suivant les efforts mesurés sur l'effecteur.

Introduction

Initialement, les robots étaient voués à un avenir prometteur grâce à une mécanique polyvalente et des commandes programmables leur conférant une flexibilité élevée. Cependant, la grande majorité des robots installés effectuent toujours des tâches routinières ("pick and place") dans un contexte de production de grande série. Cette situation est, en partie, causée par deux facteurs :

En premier lieu, l'apprentissage des positions et trajectoires requiert du temps et des modifications fréquentes en plus de mobiliser la cellule robotisée. La génération de ces positions et tracés sur un système de CFAO n'apporte qu'une solution partielle au problème. En effet, les difficultés liées au recalage de ces trajectoires et à la dynamique du mouvement subsistent quand même.

En second lieu, beaucoup de tâches industrielles usuelles requièrent un contrôle d'effort et une certaine souplesse de la part de l'ouvrier. On peut ainsi songer aux tâches d'assemblage, d'ébavurage, de ponçage, etc. Ces tâches sont actuellement difficile à faire exécuter par les robots industriels puisque ces derniers ne sont contrôlés qu'en position et en vitesse.

Dans le but d'apporter un élément de solution à ces problèmes, nous avons développé un capteur d'efforts à six degré de liberté et une interface lui permettant de communiquer avec les contrôleurs des robots de marque GMF. Pour compléter, nous avons également produit des programmes permettant le contrôle en temps réel des trajectoires en fonction des signaux provenant du capteur d'efforts.

À ce sujet, une recherche bibliographique nous a révélée qu'un certain nombre d'auteurs s'étaient intéressés à la conception de capteurs d'efforts pour la robotique [1, 2, 3, 4]. Toutefois, depuis deux ou trois années, le nombre de publications sur ce sujet a diminué de façon significative. Une des raisons de ce phénomène réside certainement dans la disponibilité de tels capteurs sur le marché (compagnies JR3 inc., Assurance Technologies inc., Schunk...). D'autre part, dans les publications aussi bien que dans le commerce, très peu de choses ont été réalisées au niveau de l'interface et de l'utilisation des signaux par un robot.

Dans cet article, nous décrivons la conception et le principe de fonctionnement d'un capteur d'efforts basé sur une géométrie offrant de nombreux avantages par rapport à ce qui existe actuellement. Le problème de l'exploitation des signaux par un robot sera également abordé.

Description générale du capteur

Le capteur d'efforts développé s'articule autour d'une géométrie du type "plate-forme de Stewart". Cette géométrie se caractérise par deux plate-formes circulaires et parallèles reliées entre elles par six membrures en traction ou compression, formant deux à deux des triangles joints en leurs sommets par des rotules. La figure 1 illustre le principe général mis en oeuvre tandis que la figure 2 montre le détail d'une membrure ou tige.

Principe de fonctionnement

Comme on peut le voir sur ces deux figures, des jauges extensométriques sont montées sur chaque tige, permettant ainsi d'en mesurer la déformation en traction ou compression. L'utilisation de rosettes et leur montage en pont de Wheatstone complet permet en outre de tenir compte du coefficient de Poisson tout en s'affranchissant des effets thermiques. D'autre part, l'utilisation de rotules pour fixer les membrures sur les supports ou plate-formes permet de s'assurer qu'elles ne subiront que des efforts axiaux générant des contraintes de traction ou de compression.

Les signaux issus des six tiges sont enfin lus de manière séquentielle par une carte d'acquisition et de traitement pour ensuite être introduits dans une matrice de découplage. Celle-ci fournit, à sa sortie, les valeurs des forces et des moments suivant les axes X, Y et Z. Le développement de cette matrice est présenté dans une section ultérieure.

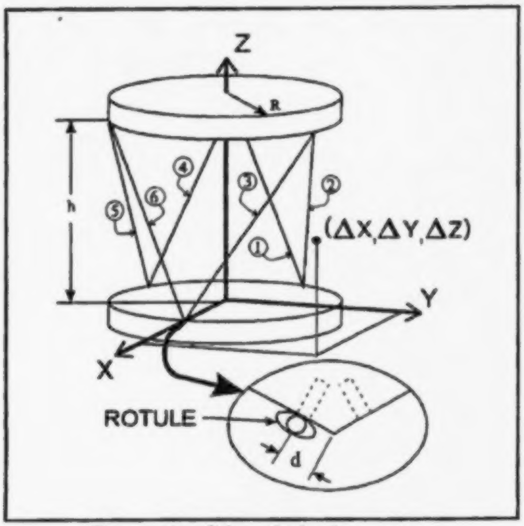


Figure 1 : Géométrie du capteur

Géométries disponibles

Au niveau de la littérature, notre recherche nous a démontrée que seuls Gaillet et Reboulet [1] se sont déjà intéressés à la géométrie "plate-forme de Stewart" pour la conception d'un capteur d'efforts. Une étude des capteurs d'efforts recensés dans le commerce, nous a en outre permis de déduire que, dans tous les cas, le principe de fonctionnement mis en oeuvre est basé sur la mesure de déformation de poutres en flexion. Un des avantages de ce genre de capteurs provient du fait qu'ils peuvent être entièrement usinés à partir d'un seul bloc. De plus, suivant l'agencement des poutres en flexion, il est possible de découpler les signaux de façon à obtenir directement les valeurs des forces suivant des axes orthogonaux. Néanmoins, pour les tâches requérant une grande précision, une matrice de découplage s'avère quand même nécessaire. Dans ce cas, cette géométrie perd évidemment son intérêt, d'autant plus que la disponibilité de moyen de calcul performant rend aujourd'hui de tels calculs relativement faciles.

Choix de la géométrie

Cela dit, la géométrie dite "plate-forme de Stewart" offre un certain nombre d'avantages sur le plan de la réalisation. Ainsi, la fabrication en est relativement facile puisque le capteur ne comprend alors que deux types de pièces simples: les deux plateaux et les six tiges. De plus, comme l'illustre la figure 2, les rotules vissées aux extrémités de chaque membrure en permettent un ajustement de la longueur ce qui simplifie évidemment l'assemblage du capteur. Enfin, il est possible d'étalonner chaque tige individuellement assurant ainsi un meilleur contrôle des défauts et une identification plus précise et plus systématique du capteur. Ajoutons également que, grâce à sa forme unique, le capteur permet d'accepter en son centre un outil tournant ou un préhenseur.

Au niveau des caractéristiques, le capteur proposé offre, comme nous le verrons, des performances respectables. En effet, l'utilisation de membrures en traction ou compression plutôt qu'en flexion présente,

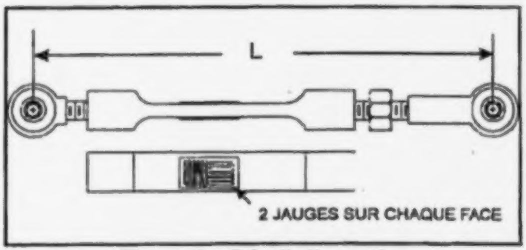


Figure 2 : Détail d'une tige

à nouveau, plusieurs avantages. Ainsi, on sait que les déformations en traction sont toujours inférieures à celles mesurées en flexion à charge équivalente. Cela permet d'obtenir, pour le capteur, une gamme de fonctionnement intéressante malgré la plage d'opération limitée des jauges de contraintes. D'autre part, pour les même raisons, cette géométrie offre également une rigidité supérieure. De ce fait, il est possible d'isoler la souplesse (compliance) du capteur permettant ainsi une analyse et un contrôle plus précis des paramètres opérationnels du capteur sur un robot.

Avant de terminer, mentionnons toutefois que les avantages présentés sont obtenus justement au détriment de déformations plus faibles. Sur le prototype réalisés, ces inconvénients sont compensés par l'utilisation de rosettes montées en pont complet ainsi que par l'intégration d'un convertisseur analogique/numérique de 16 bits. Enfin, les essais réalisés jusqu'à présent montrent que la sensibilité du capteur demeure encore assez grande. Celle-ci dépend bien sûr et surtout, de la section des membrures du capteur.

Choix des jauges

Techniquement, ajoutons également que lors de la réalisation du prototype, notre choix s'est arrêté sur l'utilisation de jauges conventionnelles plutôt qu'à semi-conducteur. En effet, bien qu'ayant un facteur de jauges beaucoup plus élevé, ces jauges sont très sensibles aux variations de température ce qui les exclue pour la plupart des applications de précision.

Modèle du capteur

Le modèle mathématique décrivant la structure proposée est connu et décrit dans Gaillet et Reboulet [1]. Toutefois, la matrice de découplage associée à cette géométrie est basée sur l'hypothèse selon laquelle les membrures sont jointes les unes aux autres à leur extrémités. Dans la réalité, tel qu'illustré dans l'encadré de la figure 1, il est impossible physiquement de les fixer en un seul point à cause de l'encombrement des rotules. Par conséquent, nous avons développé une nouvelle matrice permettant de tenir compte de cela ainsi que de la possibilité que le point d'application des forces puisse se situer ailleurs qu'à l'origine du repère du capteur.

Développement de la matrice

Le développement mathématique de cette matrice est relativement simple et repose sur la décomposition des forces mesurées dans les six tiges. Cela est fait en projetant chacune d'elle suivant les axes du repère décrit à la figure 1. Par la suite, les forces et les moments suivant les axes X, Y et Z sont obtenus en écrivant les équations d'équilibre associées. La matrice de découplage finale ne constitue, par conséquent, qu'une formulation matricielle de l'état d'équilibre du capteur. Cette matrice est présentée dans la sous-section qui suit.

Matrice de découplage

Soit, les paramètres associés au capteur :
R le rayon du capteur,
h sa hauteur,
L la longueur des membrures et,
d le décalage des rotules;
et ceux relatifs au point d'application des efforts :
ΔX le décalage suivant l'axe des X,
ΔY le décalage suivant l'axe des Y,
ΔZ le décalage suivant l'axe des Z;

La matrice finale obtenue des équations d'équilibre s'énonce comme suit:

$$\begin{bmatrix} F_X \\ F_Y \\ F_Z \\ M_X \\ M_Y \\ M_Z \end{bmatrix} = \begin{bmatrix} a & -2a & a & a & -2a & a \\ -\sqrt{3}a & 0 & \sqrt{3}a & -\sqrt{3}a & 0 & \sqrt{3}a \\ -b & -b & -b & -b & -b & -b \\ M_{X1} & M_{X2} & M_{X3} & M_{X4} & M_{X5} & M_{X6} \\ M_{Y1} & M_{Y2} & M_{Y3} & M_{Y4} & M_{Y5} & M_{Y6} \\ M_{Z1} & M_{Z2} & M_{Z3} & M_{Z4} & M_{Z5} & M_{Z6} \end{bmatrix} * \begin{bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \\ F_5 \\ F_6 \end{bmatrix} \quad (1)$$

$$\begin{aligned}
M_{X1} &= -bd\sqrt{3}/2 - b\Delta Y + \sqrt{3}a\Delta Z \\
M_{X2} &= -bR\sqrt{3}/2 - b\Delta Y \\
M_{X3} &= -b(R-d)\sqrt{3}/2 - b\Delta Y - \sqrt{3}a\Delta Z \\
M_{X4} &= b(R-d)\sqrt{3}/2 - b\Delta Y + \sqrt{3}a\Delta Z \\
M_{X5} &= bR\sqrt{3}/2 - b\Delta Y \\
M_{X6} &= bd\sqrt{3}/2 - b\Delta Y - \sqrt{3}a\Delta Z
\end{aligned}$$

$$\begin{aligned}
M_{Y1} &= b(R-d/2) + b\Delta X + a\Delta Z \\
M_{Y2} &= b(d-R/2) + b\Delta X - 2a\Delta Z \\
M_{Y3} &= -b/2(R+d) + b\Delta X + a\Delta Z \\
M_{Y4} &= -b/2(R+d) + b\Delta X + a\Delta Z \\
M_{Y5} &= b(d-R/2) + b\Delta X - 2a\Delta Z \\
M_{Y6} &= b(R-d/2) + b\Delta X + a\Delta Z
\end{aligned}$$

$$M_{Z1} = -\sqrt{3}aR - \sqrt{3}a\Delta X - a\Delta Y$$
$$M_{Z2} = \sqrt{3}aR + 2a\Delta Y$$
$$M_{Z3} = -\sqrt{3}aR + \sqrt{3}a\Delta X - a\Delta Y$$
$$M_{Z4} = \sqrt{3}aR - \sqrt{3}a\Delta X - a\Delta Y$$
$$M_{Z5} = -\sqrt{3}aR + 2a\Delta Y$$
$$M_{Z6} = \sqrt{3}aR + \sqrt{3}a\Delta X - a\Delta Y$$

avec

$$a = (R-2d)/2L\ ,\quad b = h/L\ ,\quad h = \sqrt{(L^2-(R-2d)^2)}$$

Par rapport à la matrice non compensée fournie dans [1], le résultat qui précède se distingue uniquement par la valeur des moments autour des axes X et Y (M_X et M_Y). En effet, seul ces axes sont affectés par les décalages imposés au niveau des rotules. En termes quantitatifs, cette différence varie beaucoup suivant le chargement étudié et s'avère donc significative à cet égard.

Simulation

Cela dit, la matrice de découplage peut avantageusement être exploitée pour étudier la sensibilité du capteur aux différents paramètres dimensionnels. Ainsi, avant de concevoir notre prototype, nous avons réalisé un simulateur qui nous a permis d'évaluer l'effet des dimensions sur les performances du capteur.

De fait, pour un ensemble de forces données sur les tiges, nous avons fait varier successivement le rayon R et la hauteur h du capteur et mesuré leurs effets sur la force résultante et le moment autour de l'axe Z. Nous avons alors constaté que la force était beaucoup moins sensible que le couple aux variations de ces paramètres. L'examen des résultats pour le couple nous a, de plus, permis de déduire que sa valeur admissible était proportionnelle au rayon R du capteur et inversement proportionnelle à sa hauteur h. De là, il était facile de déduire qu'il est préférable de concevoir le capteur le plus court et le plus large possible. Malheureusement, dans notre choix des dimensions finales nous avons été contraint d'une part par la grandeur des jauges et des rotules et d'autre part par notre désir de réaliser un capteur qui soit le plus compact possible.

Dimensions et caractéristiques du prototype

Dans notre cas, compte tenu de ces objectifs et de ces contraintes, nous avons retenus les valeurs dimensionnelles suivantes pour la réalisation d'un prototype:

R = 44mm, L = 110 mm et d = 7 mm
d'où on calcule h = 105 mm
avec une section de 9mm × 1,5 mm pour les tiges.

Enfin, en utilisant un aluminium traité 7075 pour fabriquer les membrures nous avons obtenus un capteur ayant les caractéristiques techniques énoncées ci-après:

Charge maximale permise sur une tige :
- Pour les jauges : 1400 N
- Pour le flambage des tiges : 2800 N

Charges individuelles maximales admissibles sur le capteur (en fonction des jauges) :

F_X	:	1160 N	M_X	:	160 N.m
F_Y	:	1120 N	M_Y	:	185 N.m
F_Z	:	8300 N	M_Z	:	90 N.m

Soulignons que ces charges maximales sont valables uniquement lorsque prises individuellement. En effet, tout cas de chargement combiné conduirait nécessairement à une réduction des charges maximales spécifiées ci-haut. Pour terminer, rappelons que la charge maximale est uniquement fonction de la section des tiges ce qui en fait un facteur facile à contrôler.

Traitement des signaux

L'acquisition et le traitement des signaux est réalisé par un module que nous avons développé à cet effet. La figure 3 illustre le schéma général du module qui comprend deux cartes distinctes. La première est analogique et sert à amplifier les signaux issus des jauges tandis que la seconde est numérique et gère l'acquisition et de traitement des valeurs.

La partie du schéma illustrant une tige présente en fait le montage des jauges en pont de Wheatstone complet. Les jauges accompagnées du symbole ε sont placées suivant l'axe des tiges pour en mesurer l'allongement tandis que celles portant l'inscription $-\vartheta\varepsilon$ sont perpendiculaires aux premières et permettent de tenir compte de l'effet de Poisson.

La carte analogique, comprend, quant à elle, une source de courant alimentant les six ponts de Wheatstone ainsi que six amplificateurs d'instrumentation reliés à chacun d'eux. Les sorties de ces amplificateurs sont ensuite raccordées à des filtres passe-bas ayant une fréquence de coupure de 20 Hz. Finalement les filtres sont eux-même branchés sur le multiplexeur de la carte numérique. Cela dit, il convient au passage de mentionner que les amplificateurs d'instrumentation contribuent à limiter la charge maximale admissible pour le capteur. En effet, ceux-ci saturent lorsque la charge dans les tiges atteint une valeur de 525 N. Cela nous donne donc un facteur de sécurité supplémentaire de 2,7 par rapport aux données techniques citées à la section précédente.

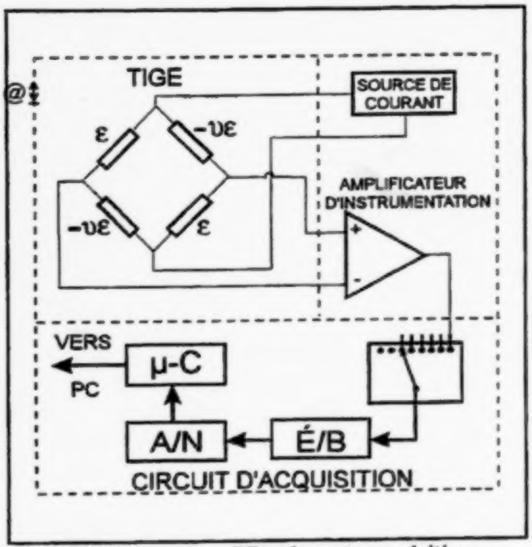


Figure 4 : Amplification et acquisition

Enfin, la carte numérique comporte, comme nous venons de le mentionner, un multiplexeur relié à un Échantillonneur/"Bloqueur" (É/B). Ce dernier joue le rôle d'une mémoire tampon entre le multiplexeur et le convertisseur Analogique/Numérique (A/N) de 16 bits.

Par la suite, le convertisseur transmets à un micro-contrôleur les valeurs numériques des signaux reçus des six tiges. Typiquement, avec les circuits réalisés, nous avons réussi à obtenir des valeurs stables à 14 voire 15 bits. Toutefois, ces performances diminuent à 12 ou 13 bits stables lorsque le capteur est monté près d'un des actionneurs du robot. Par conséquent, pour augmenter la fiabilité des résultats et diminuer l'effet du bruit, le micro-contrôleur calcule de façon continue la médiane des dix dernières valeurs lues sur chacune des tiges.

Ajoutons enfin que le micro-contrôleur assure également la gestion des différents éléments de la carte d'acquisition numérique. Ainsi, c'est lui qui synchronise les lectures du multiplexeur et du convertisseur analogique/numérique. C'est aussi le micro-contrôleur qui gère la transmission numérique des valeurs lues vers le micro-ordinateur permettant, de ce fait, leur affichage et leur communication au robot.

Intégration du capteur

Comme l'indique le titre de cet article, le but de ce travail est avant tout de réaliser un outil performant permettant le contrôle adaptatif de trajectoires robotisées. Dans cette perspective, nous avons développé les interfaces nécessaires nous permettant de faire communiquer le capteur avec des robots par le biais d'un lien série. La figure 4 montre de quelle façon le capteur et son module d'acquisition sont intégrés au fonctionnement du robot.

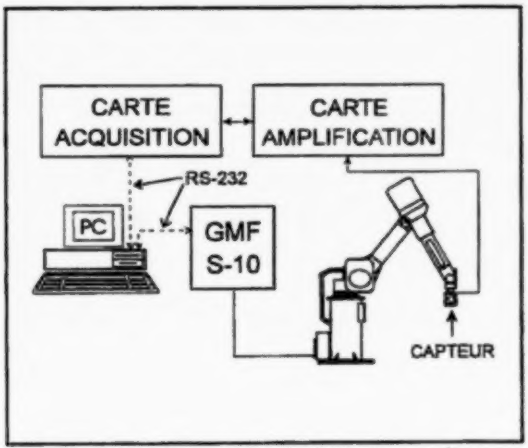


Figure 3 : Structure générale du système

Configuration du système

On voit ainsi, sur la figure 4, que la carte d'acquisition est reliée à un ordinateur personnel dont la fonction première est de calculer, en temps réel, les forces exercées sur le capteur à l'aide de la matrice de découplage présentée plus haut. Ce calcul est effectué de manière continue et les résultats sont dès lors disponibles pour être affichés et utilisés pour le calcul d'un vecteur de déplacement pour le robot.

Au niveau de la communication, c'est le contrôleur du robot qui, lorsqu'il est prêt à recevoir un ensemble de valeurs (éléments d'un vecteur), émet une requête au micro-ordinateur. Ce dernier génère alors un signal d'interruption et suspend momentanément les calculs de forces pour transmettre les éléments du dernier vecteur de déplacement calculé. Une fois les valeurs reçues, le contrôleur du robot émet l'instruction de déplacement correspondante au manipulateur. Ensuite, une nouvelle requête d'information est envoyée au micro-ordinateur. Ce cycle est répété de manière continue et c'est le programme dans le contrôleur du robot qui gère le type de tâche exécutée.

Applications en robotique

Le capteur d'efforts présenté ici trouve directement plusieurs applications en robotique industrielle. Ainsi, il est possible d'asservir les mouvements du robot de façon à conserver un effort nul sur l'effecteur. La vitesse de déplacement du robot peut alors être asservie sur la grandeur du vecteur d'efforts.

Il est également possible de contrôler les mouvements du robot pour lui faire suivre les contours d'une arête en maintenant un effort constant. Ce type de tâche constitue bien sûr une généralisation des opérations d'ébavurage, de ponçage, de meulage etc.

Travaux en cours

Dans le cas du suivi d'arête, plusieurs facteurs influent sur les performances du robot et la qualité de la trajectoire. Parmi ceux-ci, notons la souplesse (compliance) de l'effecteur et sa prise en compte dans la valeur des gains appliqués aux calculs des vecteurs de déplacement. Les types de mouvements, la précision de positionnement, la longueur des vecteurs de déplacement par rapport aux vitesses programmées sont autant de facteurs qui affectent aussi la trajectoire. De façon inverse, il semble que la forme du contour et le rayon des courbes rencontrées imposent également les valeurs de gains qu'il convient d'appliquer.

Malheureusement, aucun modèle ne nous permet d'anticiper le comportement du robot pour un ensemble de variables données. C'est pourquoi nous étudions actuellement l'effet individuel de chacun de ces facteurs sur les déplacements du manipulateur. Ces essais sont réalisés en utilisant un capteur d'efforts à un seul degré de liberté utilisant notre module d'acquisition et de traitement. Enfin, les contours utilisés demeurent dans le plan. Les résultats de cette étude feront l'objet d'un article ultérieur.

Conclusion

Nous avons présenté, dans cet article, les caractéristiques d'un capteur d'efforts à six degrés de liberté et son intégration avec des robots industriels de marque GMF pour réaliser le contrôle adaptatif de trajectoires robotisées. Jusqu'à présent, les développements réalisés sont à la hauteur de nos attentes en ce qui concerne les performances du capteur et de son électronique. Comme nous l'avons mentionné, l'intégration du capteur avec un robot a également été effectuée mais requiert encore du travail au niveau de l'optimisation de certains paramètres lors de tâches de suivi d'arêtes.

Qui plus est, de nombreuses autres applications sont actuellement en cours d'investigation. Une des plus prometteuse concerne certainement le développement d'un actionneur "intelligent". Cet actionneur serait intégré au capteur d'efforts que nous avons décrit et permettrait un contrôle local hybride force/position du robot [5, 6]. Ainsi, suivant cette configuration, le manipulateur servirait à déplacer l'effecteur de façon grossière tandis que l'actionneur "intelligent" agirait localement pour corriger la trajectoire en fonction des signaux reçus du capteur d'efforts. Un tel actionneur trouverait des applications dans un grand nombre de tâches comprenant, entre autres, l'ébavurage, le ponçage, le meulage, l'assemblage etc.

De telles applications, couplées à une intégration rapide avec les contrôleurs de robots existant élargirait évidemment la gamme d'applications des manipulateurs industriels actuellement disponibles. Pour terminer, nous croyons que cette voie est la plus prometteuse pour résoudre les problèmes liés à l'apprentissage des positions et trajectoires ainsi qu'au recalage et à la dynamique des mouvements.

Remerciements

Les travaux présentés dans cet article ont été rendu possibles grâce à des fonds de recherche provenant du :

- Programme de support institutionnel de la recherche (PSIR) et du
- Conseil de la recherche en sciences naturelles et en génie (CRSNG).

Les auteurs tiennent également à remercier Daniel Faucher et Jocelyn Lalande pour leur contribution au projet.

Références

1. A. Gaillet et C. Reboulet, « A Isostatic Six Component Force and Torque Sensor », *13th International Symposium on Industrial Robots and Robots 7 : conference proceedings*, p. 18:102-18:111, avril 1983.

2. J.R. Stubbe et E. Bayo, « Analysis and design of a six axis Truss Type of Force Sensor », *Advanced Robotics 1989. Proceedings of the 4th International Conference on Advanced Robotics*, p. 455-464, juin 1989.

3. M. Uchiyama, E. Bayo et E. Palma-Villalon, « A Systematic Design Procedure to Minimize a Performance Index for Robot Force Sensors », *Transactions of the ASME*, vol. 113, p. 388-394, septembre 1991.

4. D. Boley, « Sensor and tool development for robotic deburring », *The Industrial Robot*, p. 163-167, septembre 1987.

5. G.M. Bone, M.A. Elbestawi, R. Lingarkar et L. Liu, « Force Control For Robotic Deburring », Journal of Dynamic Systems, Measurement, and Control, vol. 113, p. 395-400, septembre 1991.

6. H. Kazerooni, « Automated Robotic Deburring Using Impedance Control », *IEEE Control Systems Magazine*, p. 21-25, février 1988.

Y05C10

Estimation of EndPoint Position and Orientation of a Flexible Link Using Strain Gauges

149691

S.J. Miller J.-C. Piedbœuf*
Mechanical Engineering Department,
Royal Military College of Canada, Kingston, Ont., Canada K7K 5L0

Abstract

This paper explains how strain gauges may be used to obtain the endpoint position and orientation of a flexible robot arm. We show that the position and orientation of the extremity of a flexible link with respect to its root may be expressed as a function of the curvature. Then, we apply an interpolation technique to obtain a continuous curvature function from a finite set of measurements made using strain gauges. Using simulated curvatures corrupted with white noise, we show that our technique is able to reconstruct the endpoint position and orientation. This is done for a static and a dynamic case. Then, we validate experimentally the static case. We found that the estimation errors of the endpoint position and orientation depended upon the number and position of the curvature measurements along the beam length.

Introduction

The topic of flexible manipulators has been studied extensively over the past few years. One of the most daunting challenges of flexible robots is the determination of the endpoint position and orientation. In a rigid robot, the endpoint position may be calculated as soon as joint positions are known. This is not the case for flexible manipulators because flexible links have their own dynamics. Up to now, external systems mainly based on vision have been used to detect the endpoint in planar motion [1]. This kind of system has the advantage to be compatible with any robot. However obstructions or interferences result in lost information. Precision can also be a problem and the analysis time is often prohibitive. Moreover, detection in 3D space has not yet been applied for real time control of flexible robots and the determination of endpoint orientation is still an unsolved problem with vision systems.

Rather than external system like vision, one may use an internal system like strain gauges. The strain gauges measure the curvatures at different points along the link. They have been used in position control to give a feedback of the link deformation and in force control as a way to measure the endpoint force [2]. Still, we have found no evidence of their use to obtain the endpoint position and orientation. Since there is a relation between the curvature and the deflection of a beam, it is certainly possible to do it. The main problem is that only a finite number of strain gauges can be used while a continuous function for the curvature is required to obtain the deflection. Therefore we must use some interpolation techniques to reconstruct the curvature along the link. Strain gauges are less prone to external interference and easily give position and orientation of any point along the link. Still, there is a calibration problem but calibration is inherent to any measurement system.

To use strain gauges to obtain the endpoint position and orientation of a flexible manipulator, we need the relation between the root of a flexible link and its extremity as a function of the curvature. It is a common assumption to consider the link's deformations as small. But it is now well accepted that in order to have the equations of motion exact to the first order, nonlinear strain-displacement relationships must be used to include the foreshortening. The foreshortening is the fact that a beam element at position x along the neutral axis in the undeformed state is no longer at position x in the deformed state. A simple way to achieve that is to use the kinematics exact to the second order [3].

In this paper, we demonstrate the use of strain gauges to obtain the endpoint position and orientation of a link flexible in only one direction. First,

---

*Assistant professor, tel: (613) 541-6309 - Fax: (613) 542-8612 - e-mail: *piedboeuf_jc@rmc.ca* also adjunct professor, Départ. de génie élec. et inf., École Polytechnique de Montréal

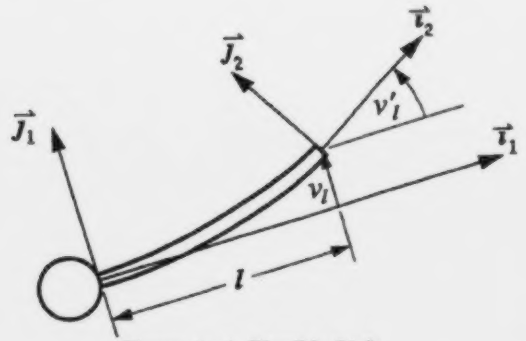


Figure 1: A Flexible Link

we develop the kinematics exact to second order in terms of the curvatures. Then we discuss the interpolation problem. To validate the approach, we simulate a static and a dynamic case. For the static case, we also have some experimental results.

Kinematic Model

In a rigid robot, the endpoint position and orientation are given using translation vectors and rotation matrices that take into account the motion of each joint and the distance between the joints. To simplify the notation, these vectors and matrices can be combined in homogenous matrices and give one transformation matrix for the endpoint, as follows:

$$\boldsymbol{T}_e = \boldsymbol{T}_1 \cdot \boldsymbol{T}_2 \cdots \boldsymbol{T}_n \tag{1}$$

The inclusion of flexible links into this scheme simply requires the insertion of appropriate transformation matrices to consider the motion of the link's endpoint with respect to its root.

In this section, we develop the kinematic model of one flexible link (Fig. 1). Specifically, we want the relation between a frame $\mathcal{R}_1$ fixed at the root of the flexible link and a frame $\mathcal{R}_2$ attached to its extremity. Our beam is only flexible in the $\vec{\imath} - \vec{\jmath}$ plane with a deflection $v = v(x,t)$ from the $\vec{\imath}_1$ axis.

To build the model, we make the following assumptions:

1. The beam is slender so the Euler-Bernoulli beam model is applicable.
2. The deformations are small ($v(l,t) < 10\% l$).
3. The neutral axis is non-extensible

We also assume a constant stiffness EI_y and a constant linear density ρ for the beam of length l. Both frames $\mathcal{R}_1$ and $\mathcal{R}_2$ have their $\vec{\imath}$ axis tangent to the neutral axis.

The orientation of frame $\mathcal{R}_2$ with respect to frame $\mathcal{R}_1$ is obtained through the rotation matrix $\boldsymbol{R}^{\mathcal{R}_1}_{\mathcal{R}_2}$ from $\mathcal{R}_2$ to $\mathcal{R}_1$. As shown in Figure 1, the link endpoint has an angle $v_l'^{1}$ with respect to the horizontal. The rotation matrix is then [4]:

$$\boldsymbol{R}^{\mathcal{R}_1}_{\mathcal{R}_2} = \begin{bmatrix} \cos v'_l & -\sin v'_l & 0 \\ \sin v'_l & \cos v'_l & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2}$$

Furthermore, assumption 2 allows us some simplifications by considering small deformations. However, as mentioned previously, the kinematics must be developed up to the second order to have a model exact to the first order. Thus, we may simplify the rotation matrix to the second order terms as:

$$\boldsymbol{R}^{\mathcal{R}_1}_{\mathcal{R}_2} \overset{\mathcal{O}(2)}{=} \begin{bmatrix} 1 - \frac{1}{2}v_l'^2 & -v'_l & 0 \\ v'_l & 1 - \frac{1}{2}v_l'^2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3}$$

where $v_l = v(l,t)$.

Yet, while this simple approach of expressing the rotation matrix in terms of the angle v' is correct for one bending displacement, it can not be extended to the case of two bending displacements and one torsion. The reason being that the second order rotation matrices are not commutative and thus depend on the order of rotation as pointed out by Hodges et al (1980) [5]. A better approach is to use the local curvatures as coordinates [3].

To express the position of the origin of $\mathcal{R}_2$ with respect to origin of $\mathcal{R}_1$, we use assumption 3 and the fact that the x axis of frame $\mathcal{R}_2$ is tangent to the neutral axis. Mathematically, this is written as follows:

$$\frac{d\,{}^1\boldsymbol{r}_{\mathcal{R}_1,\mathcal{R}_2}}{dx} = \boldsymbol{R}^{\mathcal{R}_1}_{\mathcal{R}_2} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \tag{4}$$

where the superscript 1 in ${}^1\boldsymbol{r}_{\mathcal{R}_1,\mathcal{R}_2}$ indicates that the position vector is expressed in terms of the unit vectors of frame $\mathcal{R}_1$. Replacing $(3)^2$ in (4) and solving it gives the position vector as:

$${}^1\boldsymbol{r}_{\mathcal{R}_1,\mathcal{R}_2} = \begin{bmatrix} l - \frac{1}{2}\int_0^l v'^2 \\ v_l\ 0 \end{bmatrix} \tag{5}$$

We now have the rotation matrix (which gives the orientation) and the position vector in terms of the deflection $v(x,t)$. However, since we are using

---

[1] $()'$ represents the derivative with respect to the variable x and $v'_l = v'(l,t)$.

[2] Note that $v'_l = v(l,t)$ must be replaced by $v = v(x,t)$

strain gauges, we want to express this rotation matrix and position vector in terms of the curvature $\kappa_z = \kappa_z(x,t)$. To achieve it, we use the relationship $v'' = \kappa_z$ [6]. Integrating twice we obtain:

$$v(x,t) = \int_0^x \int_0^\xi \kappa_z(\eta, t) d\eta d\xi \qquad (6)$$

Substituting (6) into (3) and (5), we obtain the rotation matrix and position vector for the end of the beam ($x = l$) in terms of the curvature:

$$R^{\mathcal{R}_1}_{\mathcal{R}_2} = \qquad (7)$$

$$\begin{bmatrix} 1 - \frac{1}{2}\left(\int_0^l \kappa_z dx\right)^2 & -\int_0^l \kappa_z dx & 0 \\ \int_0^l \kappa_z dx & 1 - \frac{1}{2}\left(\int_0^l \kappa_z dx\right)^2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$^1r_{\mathcal{R}_1,\mathcal{R}_2} = \begin{bmatrix} l - \int_0^l \frac{1}{2}\left(\int_0^x \kappa_z d\xi\right)^2 dx \\ \int_0^l \int_0^x \kappa_z d\xi dx \\ 0 \end{bmatrix} \qquad (8)$$

In equations (7) and (8), it is assumed that a function of the curvature $\kappa_z(x,t)$ is known. In practical application, since we use strain gauges, it is only possible to measure the curvature at discrete positions along the link. Therefore we must interpolate these measurements to approximate the curvature of the beam at any point.

Interpolation

For a given time t, we have a finite set of n curvatures:

$$\kappa_z = [\ \kappa_{z_1} \quad \kappa_{z_2} \quad \dots \quad \kappa_{z_n}\ ] \qquad (9)$$

with κ_{z_j} being the curvature at point x_j. From that set we want a continuous function $\kappa_z = \kappa_z(x)$ that will be valid at the given time t. The simplest interpolation is to fit the curvature κ_z to a polynomial of order N:

$$\kappa_z(x) = \sum_{j=0}^{N} a_j x^j \quad \text{where} \quad N \le n+1 \qquad (10)$$

Applying that for each of the n curvature creates the following system of linear equations:

$$\begin{bmatrix} 1 & x_1 & x_1^2 & \dots & x_1^N \\ \vdots & \vdots & \vdots & \vdots & \\ 1 & x_n & x_n^2 & \dots & x_n^N \end{bmatrix} \begin{bmatrix} a_0 \\ \vdots \\ a_N \end{bmatrix} = Xa = \kappa_z \qquad (11)$$

Note that the matrix X is of dimension $n \times (N+1)$. A polynomial of order N requires at least $N+1$ curvatures. If more curvature measurements are available ($n > N+1$), we can use the extra measurements to minimize the interpolation error. This can be accomplished through the following least squares solution [7]:

$$a = (XX^T)^{-1} X^T \kappa_z = X^* \kappa_z \qquad (12)$$

where X^* is the pseudo-inverse matrix. Since X depends only on the position of the strain gauges, which are known a priori, the pseudo-inverse X^* may be computed off-line. We then know the coefficients a_j of the polynomial as functions of the curvature measurements. We substitute (10) into (7) and (8) to obtain the rotation matrix and position vector in terms of the coefficients a_j. This calculation can be done off-line using a symbolic package such as Maple. Therefore, it is possible to express the position vector and the rotation matrix directly in terms of a finite set of curvature measurements.

Interpolation Example

To demonstrate how the beam curvature may be interpolated using the least square method, we give a small example. Consider a 1 meter beam which has strain gauges placed at positions:

$$x_1 = \frac{1}{4}\, m, \quad x_2 = \frac{1}{2}\, m, \quad x_3 = \frac{3}{4}\, m \qquad (13)$$

Let us interpolate the curvature by fitting these measurements to a polynomial of order $N = 1$.

$$\kappa_z = a_1 + a_2 x \qquad (14)$$

Substituting (13) into (11), we obtain:

$$\begin{bmatrix} 1 & \frac{1}{4} \\ 1 & \frac{1}{2} \\ 1 & \frac{3}{4} \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} \kappa_{z_1} \\ \kappa_{z_2} \\ \kappa_{z_3} \end{bmatrix} \qquad (15)$$

Solving for the coefficients a_i using (12), we get:

$$\begin{bmatrix} a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} \frac{4}{3} & \frac{1}{3} & -\frac{2}{3} \\ -2 & 0 & 2 \end{bmatrix} \begin{bmatrix} \kappa_{z_1} \\ \kappa_{z_2} \\ \kappa_{z_3} \end{bmatrix} \qquad (16)$$

Finally, we may substitute our interpolating polynomial (14) into (7) and (8) to obtain the rotation matrix and the position vector in function of the coefficients a_j:

$$R_{\mathcal{R}_2}^{\mathcal{R}_1} = \tag{17}$$

$$\begin{bmatrix} 1-\frac{l^2}{2}\left(a_1+\frac{1}{2}a_2 l\right)^2 & -l\left(a_1+\frac{1}{2}a_2 l\right) & 0 \\ l\left(a_1+\frac{1}{2}a_2 l\right) & 1-\frac{l^2}{2}\left(a_1+\frac{1}{2}a_2 l\right)^2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$^{1}r_{\mathcal{R}_1,\mathcal{R}_2} = \begin{bmatrix} l-l^3\left(\frac{1}{6}a_1^2+\frac{1}{8}a_1a_2l+\frac{1}{40}a_2^2l^2\right) \\ l^2\left(\frac{1}{2}a_1+\frac{1}{6}a_2l\right) \\ 0 \end{bmatrix} \tag{18}$$

Therefore, by taking the vector of curvature measurements $\kappa_x = [\ \kappa_{x_1} \quad \kappa_{x_2} \quad \kappa_{x_3}\ ]$ at a sample time t, we may calculate the coefficients a_j from (16) and use them in (18) and (17) to calculate the endpoint position and orientation of the beam.

Order of Interpolation

There remains the question as to which order of polynomial may best interpolate the curvature of a flexible beam from a finite set of measurements. Consider a flexible beam of length l having a distributed load ρ, endpoint force F and moment M. When the foreshortening of the beam is not of concern, the exact solution of the curvature is [6]:

$$\kappa_z(x) = -\frac{-M+F(l-x)+0.5\rho(l-x)^2}{EI_z} \tag{19}$$

Therefore, three curvature measurements would be sufficient to have the exact beam shape.

However, if we consider foreshortening or if we have a dynamic case, the situation is not as straightforward because the exact solution, when it exists, is not a simple polynomial. For example the ith mode shape of a flexible beam in vibration is described by this general equation [8]:

$$\phi_i(x) = c_1\sin\beta x + c_2\cos\beta x + c_3\sinh\beta x + c_4\cosh\beta x \tag{20}$$

We then have to find the best polynomial by trial and error. It should be noted that our polynomial is defined over the entire length of the beam. This kind of interpolation does not behave well if the order of the polynomial is too high. If a low order polynomial does not provide good results and if enough curvature measurements are available, a better option will be to divide the beam into small intervals. Then a spline technique might be used.

Simulation

To illustrate the use of the strain gauges to obtain the endpoint position and orientation for a flexible link, we develop two examples. The first one is a beam in static deformation. While this example is very simple, it is interesting because we can validate it experimentally. The second example is a rotating beam so we will consider the dynamics.

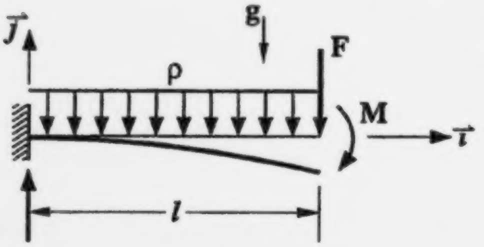


Figure 2: A flexible link in static deformation

In analyzing the simulation results we must keep some facts in mind. A typical robot has two long links which might be flexible. The error on the robot endpoint position and orientation will be the sum of the error on each link plus the error on the joints. Error on the endpoint orientation of a flexible link affects the position and the orientation of all subsequent links. For comparison, industrial robots have endpoint accuracy from a few millimeters for the large and less accurate ones, to one tenth of a millimeter for the small and more accurate ones. Also, a joint with a 500 pulses per revolution optical encoder and a 1:50 reducer has a resolution of 0.0003 radian.

Static Case

A simulation of a static flexible beam was completed to verify the validity of our model. The beam that was simulated is shown in Figure 2. In this simulation, we allowed the beam to deflect in the plane of gravity. Thus, the weight of the beam provides a constant distributed load along its length. A constant force is applied through a lever arm at the end of the beam and is always in the same direction as the gravity. This eccentric loading of the force will likewise cause a moment at the beam tip. The beam length is 0.784 m and we use a set of five strain gauges located at $x = [0.019, 0.221, 0.399, 0.589, 0.780]$ m. Furthermore, a white noise of variance 0.0005 m^{-1} was added to all curvature measurements. This noise is reflective from measurements on our experimental system (Section 5).

In the simulation, we calculated a finite set of up to 5 curvatures for a given force F which var-

Order	Δv_l [*mm*]	$\Delta v_l'$ [*rad*]
0	21	1.5×10^{-3}
1	1.3	2.7×10^{-3}
2	0.08	7.6×10^{-4}
3	0.06	7.4×10^{-4}
4	0.09	7.5×10^{-4}

Table 1: Error on predicted endpoint position (Δv_l) and orientation ($\Delta v_l'$) as function of the polynomial order for a static deformation

ied from zero to six newtons. The curvatures were then substituted into our model and used to predict the position and orientation of the endpoint. The results were then compared against the true solution.

First, we would like to show the effect of the order of the interpolating polynomial on the model. Consider the data shown in Table 1. For these simulations, all the strain gauge signals were used for the interpolation while the order was varied. Notice that the error in predicting the endpoint position decreases with an increase in the order of the interpolation. However, the error stops decreasing once the order becomes greater than 2. This comes as no surprise since we are aware that the true solution, neglecting the foreshortening, is of order 2. In comparing the errors in estimating the endpoint orientation, we notice that they are quite small and there is no improvment for orders higher than 2. The higher error for order 1 compared to order 0 is still unexplained.

Second, we are interested in determining the minimum number of strain gauges necessary for an adequate estimation of our model. Consider the data shown in Table 2. These data were calculated from simulations where the number of redundant curvature measurements was increased for increasing orders of interpolation. Notice that for orders 0 and 1, the increase in redundancy of the measurements decreased the error for both position and orientation. Still, addition of strain gauge 3 for order 0 increases slightly the position error. However, for order 2 interpolation, the error remains relatively constant. This is due to the white noise error added to the curvature measurements.

Dynamic case

For the dynamic case, we consider a beam in rotation in the horizontal plane. An eccentric payload is attached to the link extremity to generate an endpoint force and moment during motion. The beam length is again 0.808 m and we assume nine

SGset	*Ord.*	Δv_l [*mm*]	$\Delta v_l'$ [*rad*]
1	0	45.6	0.16
1, 5	0	15.6	0.013
1, 3, 5	0	19.6	0.005
1, 5	1	6.7	0.016
1, 3, 5	1	3.2	0.007
1, 2, 3, 4	1	0.62	0.003
1, 3, 5	2	0.08	0.001
1, 2, 3, 4	2	0.067	0.001

Table 2: Error on predicted endpoint position (Δv_l) and orientation ($\Delta v_l'$) as function of the polynomial order and the set of strain gauges (SG) for a static deformation

strain gauges: the first one is at $x = 0.01\ m$ and the eight others are equally spaced between $x = 0.1\ m$ and $x = 0.8\ m$. On the curvatures obtained from the simulation, we add a white noise with a standard deviation of $0.0005\ m^{-1}$. As an input to the system, we consider a cosine signal because it combines a step input at time zero that excites all the frequencies and a signal that keeps the link oscillating. This input creates a more difficult case as compared with control along a trajectory. Indeed controllers are generally done for trajectories that are smooth to avoid excitation of higher frequencies. The simulation model is linear and does not include the foreshortening effect.

With the simulation, we want to explore the effects of varying polynomial order, number of strain gauges and strain gauge locations. To study the best polynomial order, we assume that we have access to all the nine strain gauges. In Table 3, we show the average and peak errors between the endpoint position and orientation obtained from the simulation and the ones predicted using the nine curvatures. Because of the noise, mean and peak values may change slightly from one simulation to the other. Table 3 indicates that increasing the polynomial order up to 3 improves the results while orders higher than 3 do not produce any significant changes. The higher error on Δv_l for order 2 compared to order 1 is still unexplained but is probably related to the least square approach. We also note that from order 1 to 2, the error on orientation decreases by an error of magnitude while this occurs between order 2 and 3 for the position. The orientation being the derivative of the deformation with respect to x, it requires an order less than the position for a good estimation.

To evaluate the best number of strain gauges, we assume from the previous simulations that order 3

Order	Δv_l [*mm*]		$\Delta v_l'$ [*rad*]	
	mean	*peak*	*mean*	*peak*
0	10	20	0.003	0.015
1	0.5	3	0.003	0.015
2	0.7	4	0.0004	0.002
3	0.05	0.2	0.0004	0.002
4	0.05	0.2	0.0003	0.002
5	0.05	0.1	0.0004	0.002
8	0.07	0.2	0.0004	0.002

Table 3: Error on predicted endpoint position (Δv_l) and orientation ($\Delta v_l'$) as function of the polynomial order for the dynamic case

Gauges	Δv_l [*mm*]		$\Delta v_l'$ [*rad*]	
	mean	*peak*	*mean*	*peak*
1	20	50	0.08	0.2
1, 9	7	40	0.02	0.1
1, 5, 9	2	10	0.0007	0.003
1, 3, 6, 9	0.08	0.2	0.0004	0.002
1, 3, 5, 7, 9	0.08	0.2	0.0004	0.002

Table 4: Error on predicted endpoint position (Δv_l) and orientation ($\Delta v_l'$) as function of the number of strain gauges for a maximum polynomial order of 3

is sufficient. We consider sets of one to five strain gauges. Since the maximum order for the polynomial is one less than the number of strain gauges, we can only reach order 3 with sets of 4 or 5 strain gauges. Table 4 shows the average and the peak error for the endpoint position and orientation. In this table, we suppose that the gauges are numbered from 1 to 9 with 1 being the strain gauge at $x = 0.01\ m$ and 9 being the one a $x = 0.8\ m$. The sets of one, two and three strain gauges (using respectively polynomials of order 0, 1, and 2) result in large errors for the endpoint position especially for the peaks. Still, the endpoint orientation is quite good with only 3 strain gauges. Four strain gauges give very good results while five do not improve the results.

For a given polynomial order and set of strain gauges, we now discuss the best locations for the strain gauges. From previous simulations, we have shown that 4 strain gauges and polynomial order 3 gave good predictions. Now we consider 3 cases: using the first 4 strain gauges, using the last 4 strain gauges and an equal distribution of strain gauges over the length of the beam. Table 5 summarizes the results. This table indicates that the position of the strain gauge is crucial. The best approach is to distribute the strain gauges along the link.

Gauges	Δv_l [*mm*]		$\Delta v_l'$ [*rad*]	
	mean	*peak*	*mean*	*peak*
1, 2, 3, 4	2	5	0.01	0.04
6, 7, 8, 9	9	30	0.01	0.05
1, 3, 6, 9	0.08	0.2	0.0004	0.002

Table 5: Error on predicted endpoint position (Δv_l) and orientation ($\Delta v_l'$) as function of strain gauge locations for a polynomial order of 3

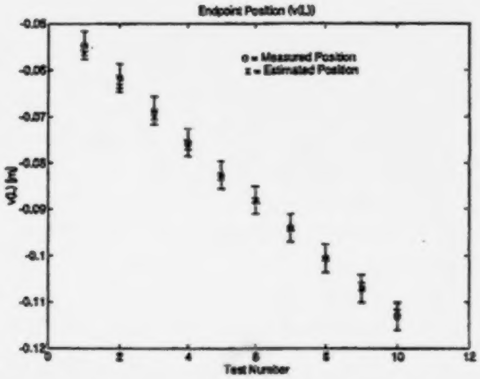


Figure 3: Measured and estimated endpoint position (v_l)

Experimental Verification

A static experiment was conducted to experimentally verify our model. The setup was identical to the one described for the static simulation (Sect. 4.1). Similarly, the force and moment at the end of the beam were created through an eccentric static load pulled by gravity. Note that the magnitude of the force and the moment do not need to be known. Only the curvature measurements from the strain gauges were needed for the model. The actual endpoint position was directly measured during the experiment. The orientaion was calculated through the use of a rigid plate attached to the end of the beam. The orientation is calculated by measuring the difference in height between the base and tip of the extension.

The results from the experiment are shown in Figures 3 and 4. The estimation of the endpoint position and orientation was achieved through the second order polynomial using gauges 1, 3 and 5. Note that the prediction of the position and orientation fall within the errors of the direct measurements.

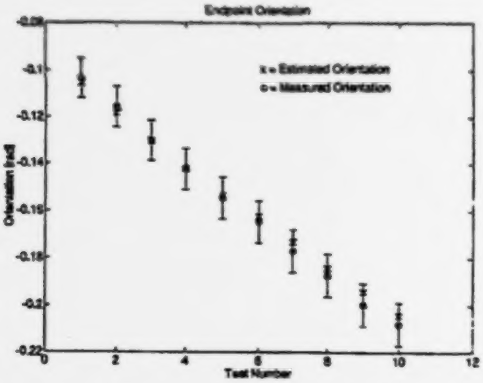


Figure 4: Measured and estimated endpoint orientation (v_1')

Conclusion

In this paper, we demonstrated the use of strain gauges to predict the endpoint position and orientation of a link flexible in one direction. As compared to vision systems, our system permits us to obtain the position and orientation of any point along a flexible link. We explained how the curvatures as measured by strain gauges are used to obtain the link endpoint position and orientation with respect to its root. Then we discussed an interpolation technique based on polynomials. Through simulations and experimentations, we have shown that the accuracy that our system can achieve is comparable to the required accuracy for industrial robots. However, the order of the polynomial and the location of the strain gauges affect the accuracy of the results. One of the main advantage of this strain gauge based sensor is its applicability for obtaining the endpoint position and orientation of a beam flexible in two directions and having some torsion.

References

[1] R. H. Cannon Jr. and E. Schmitz, "Initial experiment on the endpoint control of a flexible one-link robot," *The International Journal of Robotics Research*, vol. 3, no. 3, pp. 62–75, 1984.

[2] K. Richter and F. Pfeiffer, "A flexible link manipulator as a force measuring and controlling unit," in *1991 IEEE International Conference on Robotics and Automation*, (Sacramento, California), pp. 1214–1219, Apr. 1991.

[3] J.-C. Piedbœuf, "The Jacobian matrix for a flexible manipulator," Tech. Rep. EPM/RT-93/17, École Polytechnique de Montréal, Aug. 1993.

[4] J. Craig, *Introduction to Robotics: Mechanics and Control*. Addison-Wesley, 1989.

[5] D. H. Hodges, R. A. Ormiston, and D. A. Peters, "On the nonlinear deformation geometry of Euler-Bernoulli beams," Tech. Rep., NASA, 1980. TP 1566.

[6] J. M. Gere and S. P. Timoshenko, *Mechanics of materials*. PWS-Kent, 3rd ed., 1990.

[7] D. W. Williams and D. A. Turcic, "An inverse kinematic analysis procedure for flexible open-loop mechanisms," *Mechanism and Machine Theory*, vol. 27, no. 6, pp. 701–714, 1992.

[8] L. Meirovitch, *Analytical Methods in Vibrations*. New York: The Macmillan Company, 1967.

90C214

Stochastic Position Sensing In Unknown Environments

#149692

Terrence J. Nelson, M.Sc, B.Sc., B.E.
APRO Applied Robotics Inc., 212-112 Research Drive
PO Box 831, Saskatoon, Saskatchewan, S7K 3L7
Ph: (306) 242-2776, Fax: (306) 242-9767

Abstract

Over the last several years Applied Robotics has been developing a system to determine the position of a free roving autonomous robot in an unknown environment. The system uses data from multiple sensors and multiple readings to detect obstacles and features in the environment. Using multiple sensors and multiple readings has been found to greatly increase the accuracy and reliability of the system. The data is combined using geometric data fusion and uncertainty modeling techniques to provide a local world model. The position of obstacles and other features can be used to calculate the local position of the robot relative to these obstacles and features. The local position is relative to obstacles so it is useful in path planning and obstacle avoidance. The system was implemented using 2 cameras and 2 wheel encoders as the sensors for testing in an indoor environment. The cameras provided stereoscopic machine vision and the wheel encoders provided simple dead reckoning. By using this simple sensor suite the system achieved a position accuracy of approximately 9 mm while it moved in an unknown laboratory environment. Applied Robotics is currently planning to install the Stochastic Positioning System onto the ANCAEUS unmanned robotics vehicle system and field testing the positioning system to determine the accuracy and feasibility for use in military applications.

Introduction

There are many military and commercial applications where it is cost prohibitive or dangerous to have a human perform a particular task [1-7]. To perform these tasks a mobile robot capable of navigating in an unknown environment will most likely be required. To navigate the robot will require a position sensing system to provide the current position and orientation.

Applied Robotics has been investigating the use of multiple sensor systems to produce low cost position sensing systems. The objective of the research was to implement a low cost position sensing system that could accurately follow a given trajectory in an unknown environment and produce a map of the environment. Current research involves the use of video cameras and wheel encoders. These sensors were chosen because the video cameras are often already mounted on mobile robots, and the wheel encoders are extremely low cost.

Stochastic systems were used because they often provide a more cost effective, accurate, reliable, and robust measurement using several low cost sensors than a single high cost sensor.

The Stochastic Position Sensing System detects naturally occurring stationary features in the environment and uses these features as positional references. The position of these features can be estimated relative to the robot coordinate system. If the original robot position is selected as the origin of the world coordinate system then the positional references are estimated in world coordinates. As the robot moves the positioning system can estimate the position of the robot relative to these positional references and update the estimate of the positional references.

A position sensing system could detect naturally occurring, stationary, positional references in the environment and determine the robot's position relative to these references. There are often more positional references available than required to determine the position, this redundant data can be used to statistically improve the accuracy and robustness of the positioning sensing system.

Stochastic Systems

A Stochastic System is a system that deals with events that develop in time or space and which cannot be described precisely except in terms of probability theory [8]. Variables in a Stochastic System are random variables that can be represented by probability density functions (pdf) and are referred to as stochastic variables.

The Central Limit Theorem guarantees that, independent of the actual distributions, the system can be modeled with a gaussian distribution if the number of independent random variables is large [9, 10]. For smaller sample sizes the actual distribution can be modeled and the difference between the model and the measured samples can be approximated as gaussian [11]. This paper assumes the random sensor errors can be modeled as gaussian distributions.

The advantage of assuming a gaussian distribution is the stochastic variable can be completely represented by the mean and variance. The Stochastic Variables can be represented using Uncertain Geometry. [12]. The mean values can be represented as a vector [x1, x2, x3, ... , xn]^T, and the variance can be represented as a 2 dimensional covariance matrix:

$$V = \begin{bmatrix} v[1,1], v[1,2], \ldots, v[1,n] \\ v[2,1], v[2,2], \ldots, v[2,n] \\ \\ v[n,1], v[n,2], \ldots, v[n,n] \end{bmatrix} \tag{1}$$

where:
v[i,i] is the variance of parameter i.
v[i,j] is the covariance between parameter i
and parameter j.

The uncertainty can also be represented using the Information Matrix 'I' rather than the Covariance Matrix 'V'. The Information matrix is the inverse of the Covariance Matrix.

$$I = V^{-1} \tag{2}$$

When representing partial data the Covariance Matrix has infinite uncertainty for the parameters not represented. The handling of infinity is difficult when using a finite computer therefore the uncertainty is poorly represented. The Information Matrix is better for representing partial data because the Information Matrix will have zeroes in the matrix to represent zero information about the uncertainty rather than infinity to represent infinite uncertainty. This makes the Information Matrix easier to handle and process.

Sensor Fusion

Sensor Fusion is the combination of multiple sensor data in a stochastic fashion to obtain a higher degree of accuracy in position than using the sensors individually. Multi-sensor fusion has been used in many position sensing applications [13 - 15].

The sensor data can be multiple readings of the same sensor, multiple sensors of the same type, multiple sensors of different types, or any combination of these. These multiple sensors can be cooperative, complementary, or competitive depending on the nature of the sensors. Cooperative sensors provide information that is required by another sensor. Complementary sensors provide information that the other sensor could not measure. Competitive sensors provide the same, and possibly contradicting, information.

Sensor Systems

Every sensor can be modeled by it's ideal behavior, called an ideal sensor model. Of course in the real world sensors don't operate ideally. Some of the noise sources in the sensor can be accounted for and a correction applied; these are referred to as Deterministic Errors. An example would be the temperature effects on the bias of a sensor, wind on a sonar ranger, or camera orientation on a vision system. There are also noise sources that are random that can't be removed from the data; these are referred to as Random Errors. Examples of random errors are thermal noise, and quantization noise. In this paper random noise sources are modeled as gaussian. The sensor output can be represented as:

$$\mathit{SensorData} = \mathit{Ideal\ Sensor} + \mathit{Deterministic\ Error} + \mathit{Random\ Error}. \tag{3}$$

The mean of the Sensor Data can be used to calculate the idealized output by removing the Deterministic Errors. The standard deviation of the Sensor Data represents the Random Error and can be used to calculate the quality of the data from the sensor. A small standard deviation means the sensor data is highly accurate, a large standard deviation means the sensor data is highly uncertain.

This paper will discuss the determination of the position of a mobile robot in uncertain environments using two wheel encoders and two video cameras.

Wheel Encoder System

The Wheel Encoder System monitors the robot's position and orientation using wheel encoders to sense the displacement of the left and right wheels. The wheel encoders are rotary position encoders placed on the axles of the two drive wheels. They are capable of determining direction and angular displacement of the axle. The wheel encoders output an incremental wheel position in the form delta-x, delta-y, delta-theta. Delta-theta is the rotational transformation from the previous robot coordinate frame, and delta-x and delta-y are the translation from the previous robot coordinate frame.

Wheel encoders are prone to slippage so their positional data is inaccurate over the long term. However, since the slippage coefficient can be estimated as a deterministic error, the wheel encoders provide good short term data and can be useful in maintaining world model consistency.

Machine Vision System

The Machine Vision System uses two video cameras to estimate the position of vertical edges in the environment relative to the robot. The video cameras are passive photon detectors used to measure the light intensity of a scene. The video cameras are mounted as a stereo pair that can scan independently. The video cameras scan about the centre of the camera base and produce a 360 degree scan for each of the cameras. The camera produces an image (640V x 480H) with a viewing angle of 30 degrees. The scan is generated by taking a video frame every 30 degrees and combining 12 images into a 360' scan. Typical left and right video images are shown in Fig. 1 and Fig. 2.

The Video Images are pre-filtered using a horizontal Gaussian Filter [16, 17] and then processed using a Sobel vertical edge detector [16, 18] to produce Gradient Images.

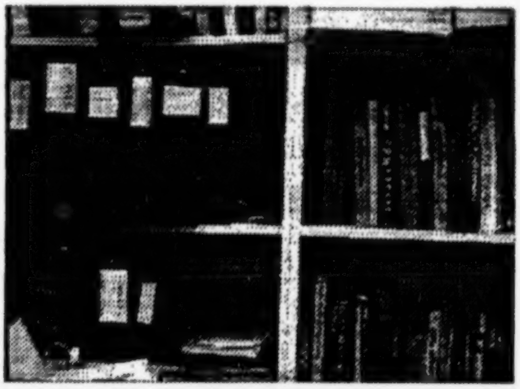


FIGURE 1: Left Video Image

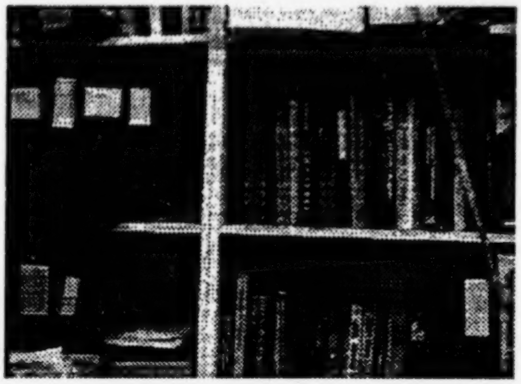


FIGURE 2: Right Video Image

The pixels in the Gradient images are summed horizontally to produce an Edge Plot. The Edge Plots provide a fast and effective representation of vertical edge gradients versus angular position in the Video Images. Typical left and right edge plots are shown in Fig. 3 and Fig. 4.

The strongest peaks in the Edge Plots are used to calculate the position of vertical edges relative to the robot. The position of the edges is uncertain so the Edges are represented by stochastic variables.

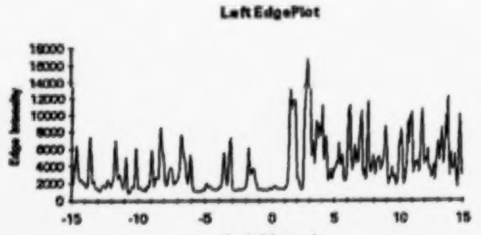


FIGURE 3: Left Edge Plot

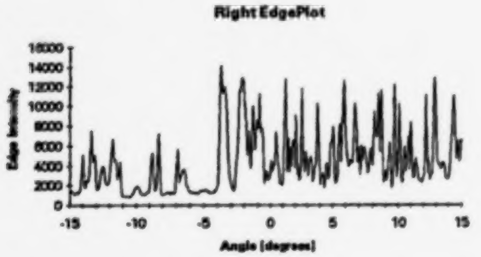


FIGURE 4: Right Edge Plot

World Modeling

The World Model is a representation of the world as seen by the robot. The structure of the World Model is application specific and depends on what sensors acquire the data, how the data is stored, indexed, referenced, and how the robot makes use of the data.

In this investigation the World Model was represented by the uncertain position of vertical edges relative to a fixed World Coordinate System. The environment is unknown so the origin of the World Coordinate System is arbitrarily set to the initial robot position. The edge positions are uncertain so they will be represented by stochastic variables.

The edges in the World Model were estimated by fusing data from the wheel encoders and the vision system. The vision system provides accurate relative angular positions of vertical edges but the machine vision system has problems with stereoscopic mismatching and range estimation. The wheel encoders provide good short term displacement data but wheel slippage produces long term drift and some short term gross errors. The wheel encoder is useful in resolving stereoscopic mismatches and can improve the range estimation by effectively increasing the baseline for the angular triangulation. The measurement of edges by the vision system is relatively independent of wheel slippage, allowing the slippage of the wheel encoders to be estimated.

Fig. 5 shows a World Model that was experimentally derived by a robot in an uncertain environment. The environment was a laboratory at Applied Robotics.

Position System

The Position System estimates the robot's position and orientation.. There are many types of position systems which can be separated into 5 generalized classes, Dead-Reckoning, Fixed Reference, Apriori Map Reader, Automatic Map Generation, and Stochastic Positioning. The Stochastic system recognizes the uncertain nature of sensor data and the world model [19]. The position can be determined from the relative location of features in the real world. Since the position of these features is uncertain then the estimated robot's position will also be uncertain.

In this application the robot position and orientation are estimated by calculating possible robot positions using triangulation of robot targets and edges in the world model. The wheel encoder system is used to reject improbable positions. The remaining possible robot positions are used to estimate the robot's position and orientation.

In testing, the Stochastic Position Sensing System successfully estimated the robot's position along a trajectory. The errors were in the range from 4.5 mm to 11 mm, with the average error of 9 mm. The errors appeared to be a random distribution centered on the actual trajectory.

Conclusions

The Stochastic Positioning System provided a cost effective, accurate, reliable, and robust positional estimate using several low cost sensors. By using sensors that are commonly already available on mobile robots (video cameras and wheel encoders),

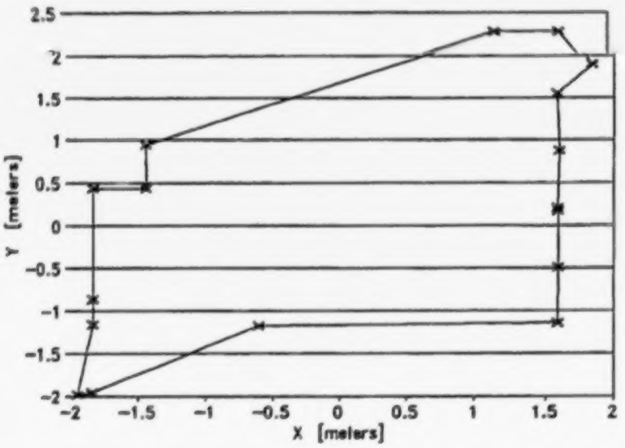


FIGURE 5: World Model

the positioning system can be incorporated without additional sensors. The system demonstrated that it could detect features in an unknown enviroment and create a model of the environment. These features then used positional references to navigate in the unknown environment. By using stochastic variables to represent the position of features in the model and the position of the robot, allowed the system to reject gross errors common to stereoscopic vision and wheel encoder systems. The robot position was accurately estimated by using the redundant sensor data.

References

1. Kwitowski, A.J.; Lewis W.H.; and Mayerchech W.D.; "Computer-Based, Tele-Operation of a New Highwall Mining System", *Proc. of 1989 IEEE Intl. Conf. on Robotics and Automation*, Vol. 3, pp. 1478.

2. The Northern Miner; *Mining Explained, a guide to prospecting and mining*, John S. Cooke, 1990.

3. Pang, Grantham K.H.; Shen, Helen; "Intelligent Control of an Autonomous Mobile Robot in a Hazardous Material Spill Accident-a Blackboard Structure Approach"; *Robotics and Autonomous Systems*, pp. 351-365, 1990.

4. Markov, Alex B.; Herring, Robert W.; "HATT-X: A High Performance Flying Laboratory for Autonomous Unmanned Aircraft R&D", *3rd Conference on Military Robotic Applications*, pp. 136-142, 1991.

5. Swinson M.L., Major U.S. Army; and Hennebeck, Major USMC; "U.S. Tactical Unmanned Ground Vehicle Project", *3rd Conference on Military Robotic Applications*, 1991, pp. 12.

6. Light, Russel D.; and Morison, Jamie; "The Autonomous Conductivity-Temperature Vehicle: First in the Seashuttle Family of Autonomous Underwater Vehicles for Scientific Payloads", *Oceans '89*, Sept. 1989, pp. 793.

7. Lescoe, Paul; Lavery, David; Bedard, Roger; "Navigation of Military and Space Unmanned Ground Vehicles in Unstructured Terrains", *3rd Conference on Military Robotic Applications*, 1991, pp. 61.

8. Ralson, Anthony; etal.; *Encyclopedia of Computer Science and Engineering*, 2nd Ed.; pp. 1409; Van Nostrand Reinhold Company Ltd.; Scarbourgh, Ont., Canada.

9. Kennedy, John B.; Neville, Adam M.; *Basic Statistical Methods for Engineers and Scientists*, IEP, New York; 1976; 2nd Ed.

10. Bethea, Robert M.; Duran, Benjamin S.; Boullion, Thomas L.; *Statistical Methods for Engineers and Scientists*, Decker, New York; 1985; 2nd Ed.

11. Jazwinski, Andrew H.; *Stochastic Processes and Filtering Theory*, Academic Press; 1970.

12. Durrant-Whyte, H.F.; *Integration, Coordination and Control of Multi-Sensor Robot Systems*, Kluwer Acedemic Publishers; 1988.

13. Abidi, M.A.; Gonzalez, R.C.; "The Use of Multisensor Data for Robotic Applications", *IEEE Transactions of Robotics and Automation*, vol. 6, No.2, April 1990.

14. Everett, H.R.; "A Programmable Near-Infrared Proximity Detector for Robot Navigation"; *SPIE Vol. 727 Mobile Robots*, 1986.

15. Durrant-Whyte, H.F.; "Consistent Integration and Propagation of Disparate Sensor Observations", *Intl. J. Robotics Research*, Vol. 6, No. 3; 1987.

16. Levine, Martin D.; *Vision in Man and Machine*, McGraw-Hill, ISBN 0070374465, 1985.

17. Wells, William M. III; "Efficient Synthesis of Gaussian Filters by Cascaded Uniform Filters", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI 8, No. 2, 1986.

18. Burt, Peter J., "Fast Algorithms for Estimating Local Image Properties", *Computer Vision, Graphics, and Image Processing 21*, pp. 368-382, 1983.

19. Ayache, Nicholas; Faugeras, Oliver D.; "Building, Registration, and Fusing Noisy Visual Maps"; *Intl. J. Robotics Research*, Vol. 7, No. 6; 1988.

414959

Model-Based Visual Recognition of 3-D Objects Using Pseudo-Random Grid Encoding

N. Trif, E.M. Petriu
University of Ottawa, E.E. Dept.
Ottawa, Ont., K1N 6N5, Canada

W.S. McMath, S.K. Yeung
Canadian Space Agency
Ottawa, Ont., K2H 8S2, Canada

#149693

Abstract

This paper discusses a grid node identification method by "pseudo-random binary array" (PRBA) encoding and its application to 3-D model-based object recognition.

Introduction

The general 3-D object recognition paradigm was defined by Besl and Jain, [1], as follows:

"Given:
- a collection of labeled solid objects in which each object may be examined and labeled models may be created using information from this examination,
- an image corresponding to one particular, but arbitrary, field of view of the real world,
- a database of previously defined set of objects.

Determine:
- what object appears in a given image,
- the 3-D location and orientation of that object with respect to a known coordinate system."

Grid encoding represents a versatile machine vision technique for this object recognition problem, [2] and [3]. Ambiguities may occur, however, because, due to image perspective errors, the grid nodes on the object surface seen in the 2-D image are no longer arranged in orderly fashion as in the encoding grid. The usual way to solve this problem is to establish an *a priori* correspondence between the detected nodes and their initial position in the encoding grid [4]-[7].

This paper discusses an object recognition method using grid node identification by PRBA encoding. All visible surfaces of the objects of interest are permanently encoded (using distinct graphical symbols) with the terms of a PRBA. The window property allows identification of the window's absolute coordinates within the PRBA from a relatively small subset of symbols recognized on the object surface using monocular vision. Knowing how different surfaces of the geometric models of the encoded objects were originally mapped on the encoding array, it is then possible to recover the identity and the "position, orientation, and scale estimation" (POSE) parameters of the object containing the recovered PRBA window by a simple consultation of the data base storing this mapping.

Pseudo-Random Grid Encoding

A pseudo-random binary array $\{A(i,j) \mid i = 0,1,2,\ldots,n1-1;\ j = 0,1,\ldots,n2-1\}$ can be obtained as shown in Fig. 1, by folding a (2^n-1) -term pseudo-random binary sequence $\{S(p) \mid p = 0,1,\ldots,2^n-2\}$ generated by a modulo-two n-bit shift register. The following relations hold, [8]:

$$2^n-1 = 2^{k1 \cdot k2}-1$$

$$n1 = 2^{k1}-1$$

$$n2 = (2^n-1)/n1$$

where *n1* and *n2* must be relatively prime.

The 2-D coordinate recovery is based on the PRBA window property. According to this, any binary pattern seen through a k1-by-k2 window sliding over the array is unique and may fully identify the window's absolute coordinates (i, j) within the PRBA. The "pseudo-random / natural" code conversion is implemented as a memory stored table. More details on this PRBA encoding are given in [9].

Two distinct graphical symbols, illustrated in Fig. 2, have been used to mark the binary values "1" and "0" within the PRBA. The particular shapes of these binary symbols were selected in such a way as to meet the following requirements: (a) there is enough information at the symbol level to provide an immediate indication of the array's orientation and about where neighboring symbols can be found; (b) the symbol recognition procedure is invariant to position, orientation, scaling, and perspective transformations; (c) the symbols should have sufficient uniqueness so that other objects in the scene won't be mistaken for encoding symbols.

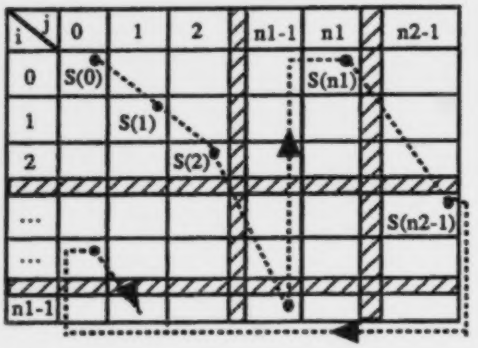


Fig. 1: A (2^n-1)-bit PRBS $\{S(p) \mid p = 0,1, ..., 2^n-2\}$ is folded to produce an $n1$-by-$n2$ PRBA

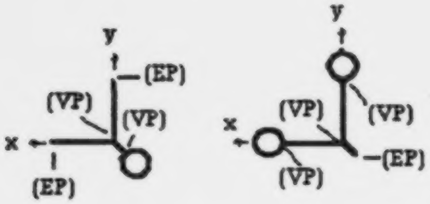


Fig. 2: The two graphical symbols used to mark "0" and "1" within the PRBA. The EP notation identifies an end point and VP identifies a vertex in the skeleton of these symbols.

The encoding symbols are recognized by their numbers of end-points and vertices in the skeletonized binary image. The symbol representing "0" has 2 end-points and 2 vertex-points and the symbol for "1" has 1 end-point and 3 vertex-points. The symbol recognition algorithm starts with a linear scan of the skeleton image for vertices. This is an efficient approach, as skeletons that don't contain vertices are rejected immediately. This excludes the vast majority of noise from being evaluated. The PRBA reconstruction procedure is independent of the employed symbol shapes and the corresponding symbol recognition technique.

Object Recognition

The visible surfaces of each object of interest are permanently encoded with elements of a large PRBA in such a way that any region of the array is used only one time. This encoding has the notable advantage of using only one bit per grid step, independent of the desired grid resolution.

Using polyhedral geometric models of the 3-D objects the object surfaces are flattened and then unfold, as illustrated in Fig. 3, to be mapped on the encoding PRBA.

The mapping of the models on the PRBA is implemented as a relational database using a 20 data element dictionary: ***object_id, object_name,*** *number_of_faces, number_of_vertices, number_of_edges,* ***face_id,*** *face_name, number_of_corners, number_of_lines, i_org, j_org,* ***corner_id,*** *i_start, j_start, i_end, j_end,* ***line_id,*** *length_of_line, i_pos, j_pos.*, (where the elements in bold characters are part of primary keys of the database relations).

There is only one polygonal face which may contain a given point A(i,j). The database system will report the *object_name*, its *object_id* attribute, and the *face_id* of the object where a given point is located. The system makes it possible to permit to find this face and also the position on this face of the given point.

The identification of the visualized object surface and its POSE recovery is illustrated in Fig. 4. Usually, only part of the encoding symbols marked on the visualized object surface are recognized. Using these symbols, a broken portion of the PRBA is reconstructed and then inspected until a complete k1-by-k2 window is found. The "pseudo-random/natural" code conversion of the recovered window contents yields the (i,j) coordinates of the origin P(i,j) of that window within the encoding PRBA.

Consulting the mapping data base, it is possible to recognize the specific polygonal object face (and thus the object) containing the recovered window. A simple algorithm allows to decide if a recovered window of origin P(i,j) belongs or not to a given convex polygon: if the sum of the areas of all triangles formed by the P(i,j) and each edge of an object face is equal to the area of the same object face, then the P(i,j) belongs to this face; if the sum of these areas is greater then the area of the object face then P(i,j) does not belong to this face.

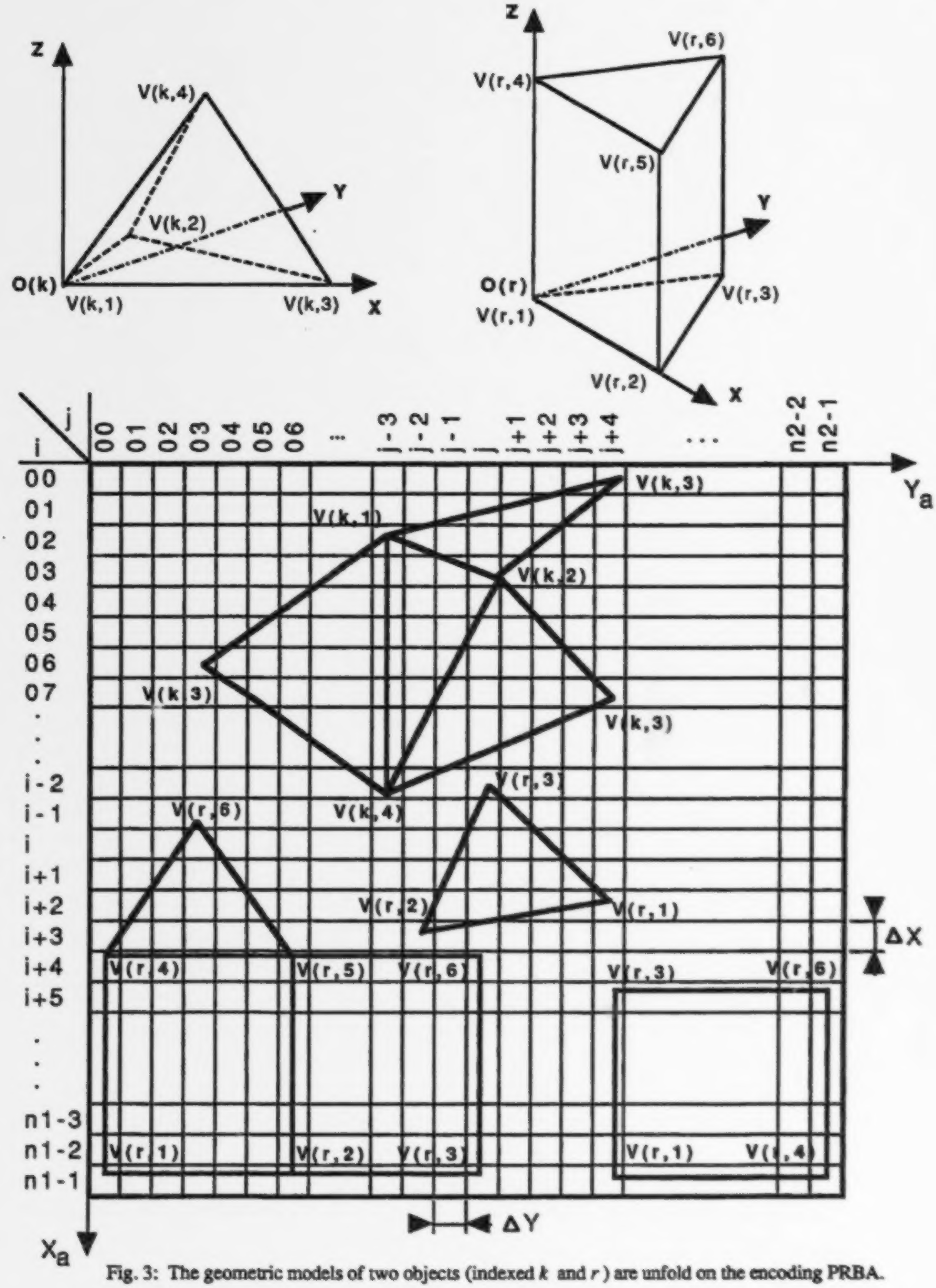

Z
V(k,4)
V(k,2)
Y
O(k)
V(k,1)
V(k,3)
X
Z
V(r,6)
V(r,4)
V(r,5)
Y
O(r)
V(r,1)
V(r,3)
V(r,2)
X
j
i
00 01 02 03 04 05 06 ... j-3 j-2 j-1 j j+1 j+2 j+3 j+4 ... n2-2 n2-1
Ya
00 01 02 03 04 05 06 07 . . . i-2 i-1 i i+1 i+2 i+3 i+4 i+5 . . . n1-3 n1-2 n1-1
V(k,3)
V(k,1)
V(k,2)
V(k,3)
V(k,3)
V(k,4)
V(r,3)
V(r,6)
V(r,2)
V(r,1)
ΔX
V(r,4)
V(r,5)
V(r,6)
V(r,3)
V(r,6)
V(r,1)
V(r,2)
V(r,3)
V(r,1)
V(r,4)
ΔY
Xa

Fig. 3: The geometric models of two objects (indexed *k* and *r*) are unfold on the encoding PRBA.

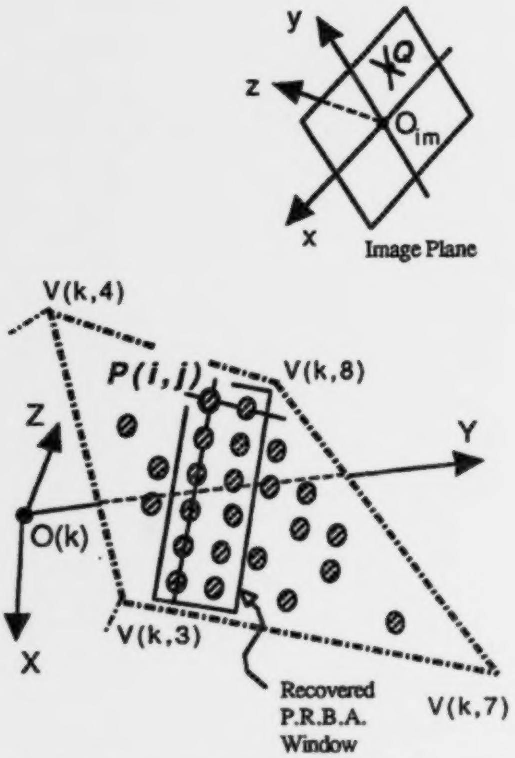


Fig. 4: The two reference systems, of the object model and the image, involved in the POSE measurement problem.

POSE Parameters Recovery

Once an object face is identified, the measurement of its POSE parameters is treated as a camera calibration problem, [10] and [11].

The POSE problem, also known as the camera calibration problem, [11], or image-to-database correspondence problem, [12], can be stated as follows: "Given a set of *m* control points (the origin of *m* symbols in the grid used to encode a face of an object), whose 3-D coordinates are known in the object's model frame, and given an image in which these *m* points are visible, determine the location of the camera (relative to the object's model frame) from where the image was taken."

Let us consider *m* control points $\{Q_i(X_i, Y_i, Z_i) \mid i=1,\dots, m\}$ whose coordinates are known in the object's model frame. Corresponding to each point Q_i there is a point $P_i(x_i, y_i, 0)$ in the image plain.

A unique solution for the POSE problem can be obtained using a minimum of six arbitrary control points. It requires to determine the 12 coefficients of the 3-by-4 homogeneous matrix that specifies the mapping from the 3-D object's model frame to 2-D image frame:

$$[tx_{P_i}, tx_{P_i}, t]^T = \begin{matrix} C_{11} & C_{12} & C_{13} & C_{14} \\ C_{21} & C_{22} & C_{23} & C_{24} \\ C_{31} & C_{32} & C_{33} & C_{34} \end{matrix} \cdot [X_{Q_i}, Y_{Q_i}, Z_{Q_i}, 1]^T.$$

for $i=1,\dots,6$.

For six control points a system of 18 linear equations has to be solved to find the camera calibration parameters. Such an approach is not suitable because it is computationally expensive.

A simpler analytical solution for the POSE problem, called the "Perspective-4-Point Problem" (P4P), can be obtained for a minimum of four control points lying in a common plane such as a face of a polyhedral object. The P4P problem, illustrated in Fig. 5, can be stated as follows, [12]:

"Given:
- a correspondence between four points $\{P_i(X_i, Y_i, 0) \mid i=1,\dots,4)$ lying in a plane in 3-D space (the object plane in our case), and four points $\{Q_i (x_i, y_i, 0) \mid i=1,\dots,4\}$ lying in the image plane,
- the distance between the center of perspective (CP) and the image plane(the focal length *f* of the image system),
- the principal point of the image plane (PPI).

Find the location of the center of perspective CP relative to the coordinate system of the object plane."

The following algorithm has been used to solve this P4P problem, [13]:

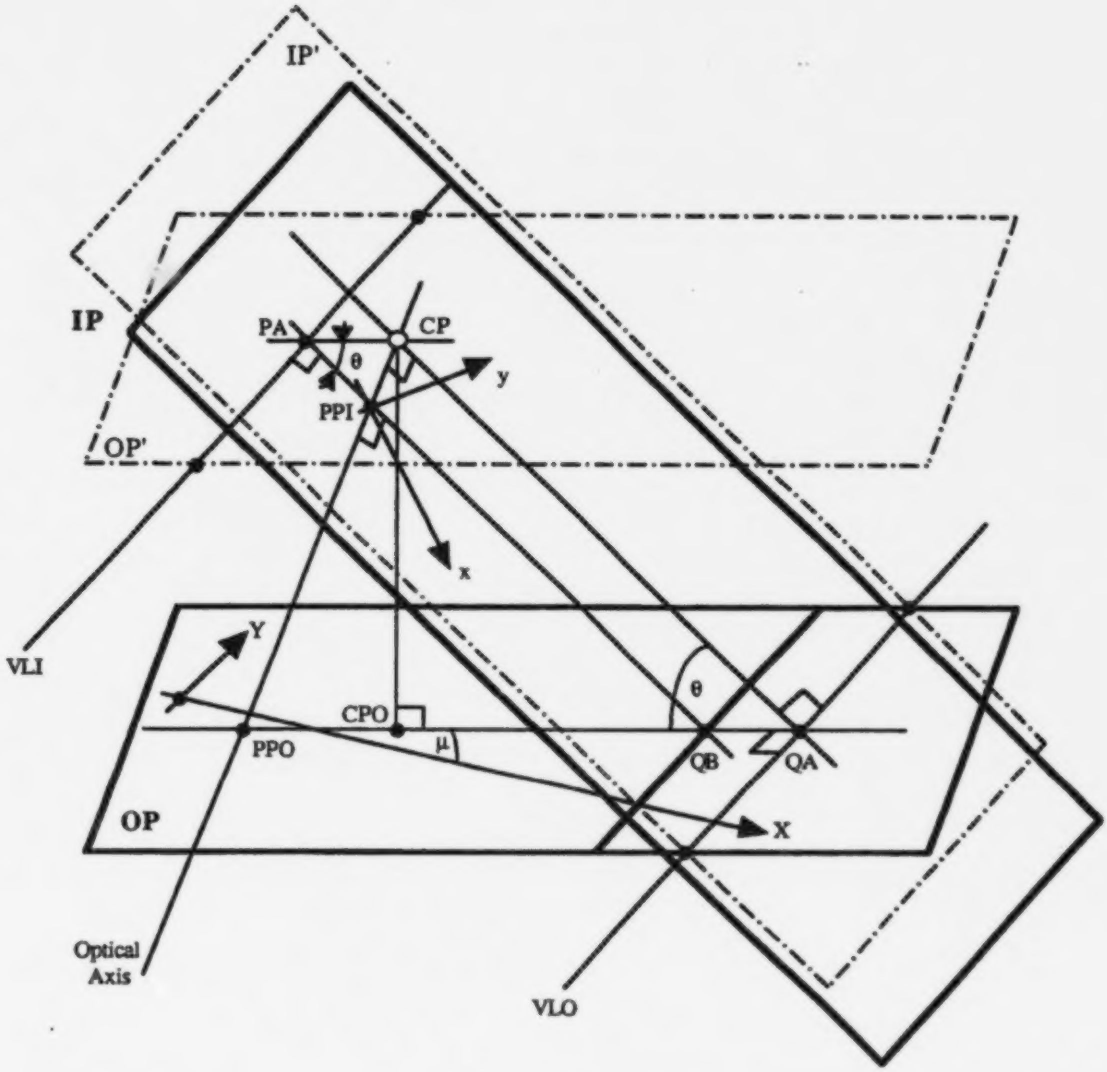
1) Determine the T[3][3] collineation matrix which maps points from the object plane to image plane: [Pi]=[T]·[Qi].

2) Determine the vanishing line in the image plane (VLI) by mapping the ideal line in the object plane [0, 0, 1] using the transformation:

   $[VLI]=[\mathrm{inv}\ [T]]' * [0, 0, 1]' = [a_1, a_2, a_3]$.

3) Determine the distance DI from PPI to the vanishing line (VLI): $DI = a_3 / \sqrt{a_1^2 + a_2^2}$.

4) Determine the tilt angle between the image plane and object plane: $\mu = \mathrm{arctg}\ (f/DI)$.


IP'
IP
PA
CP
θ
y
PPI
OP'
x
VLI
Y
CPO
θ
PPO
μ
QB
QA
OP
X
Optical Axis
VLO

IP =Image Plane
OP =Object Plane
VLI =Vanishing Line in Image plane
VLO =Vanishing Line in Object plane
CP =Center of Perspective
CPO =Center of Perspective projection in Object plane
PPI =Principal Point in Image plane
PPO. =Principal Point in Object plane

Fig. 5: Geometry of the P4P problem.

5) Determine the vanishing line in the object plane (VLO) by mapping the ideal line in the image plane [0, 0, 1] using the transformation:

   $[VLO]= [T]' \cdot [0, 0, 1]' =[b_1, b_2, b_3]$.

6) Determine the location of point (PPO) where the optical axis of the lens pierces the object plane:

   $[PPO]= [inv\ [T]]' \cdot [0, 0, 1]' =[c_1, c_2, c_3]$.

7) Determine the distance from PPO to the VLO:

   $DO=(b_1 \cdot c_1+b_2 \cdot c_2+b_3 \cdot c_3)/(c_3 \cdot \sqrt{b_1^2+b_2^2})$.

8) Determine the pan angle between the normal to $VLO=[b_1,b_2,b_3]$ and the X axis in the object plane: $\theta=arctg\ (-b_2/b_1)$.

9) Determine XSGN and YSGN. If a line (parallel to the X axis in the object plane) through PPO intersects VLO to the right of PPO, then XSGN=1. Otherwise XSGN=-1. Similarly for YSGN. Thus:

if $(b_1 \cdot c_1+b_2 \cdot c_2+b_3 \cdot c_3) / (b_1 \cdot c_3) <0$ then XSGN=1, otherwise XSGN=-1

if $(b_1 \cdot c_1+b_2 \cdot c_2+b_3 \cdot c_3) / (b_2 \cdot c_3) <0$ then YSGN=1, otherwise YSGN=-1.

10) Determine the location of the center of perspective CP(XCP,YCP,ZCP) in the reference system of the object plane:

$DCP=DO \cdot \sin \mu$

$XCP=XSGN \cdot abs\ (DCP \cdot \sin \mu \cdot \cos \theta) + c_1/c_3$

$YCP=YSGN \cdot abs\ (DCP \cdot \sin \mu \cdot \sin \theta) + c_2/c_3$

$ZCP=DCP \cdot \cos \mu$.

Experimental Results

Four representative types of 3-D objects shown in Fig. 6 were considered: (a) rectangular parallelepiped, (b) triangular prism, (c) square pyramid, and (d) right circular cylinder. Each encoded face of these objects was vertex mapped in a 63-by-65 PRBA as illustrated in Fig. 7 for a triangular prism.

Experimental results shown in Fig.8 for the triangular prism named "PrismH3L2" illustrate the performance of the discussed PRBA grid encoding method for visual recognition of 3-D objects.

The vision system uses a CCD camera and a "Matrox PIP 512" image acquisition board resident in a 286 personal computer with a 8 Mhz clock rate requiring a processing time of 18 s per image.

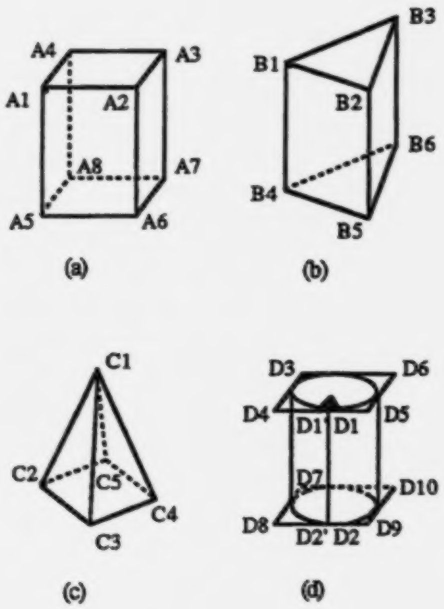


Fig. 6: Geometric model definition of 3-D objects:
(a) rectangular parallelepiped,
(b) triangular prism, (c) square pyramid, and
(d) right circular cylinder.

The described method appears to be robust in the sense that it never produced a grid with false PRBA patterns. Symbols may not be recognized, or a "1" symbol may be recognized as a "0" symbol due to a break in the skeleton and sometimes noise may be recognized as a symbol. However, during testing, these falsely recognized symbols never had the parameters to fit to their neighbors as required for acceptance and integration into the PRBA grid. The reconstructed grid may thus be incomplete, however the information which it contains is still reliable.

Acknowledgement

This work was funded in part by the Canadian Space Agency under the Contract 9F009-0-4097/00 and by the NSERC of Canada.

References

[1] Besl P. J., Jain R. C., "Three-Dimensional Object Recognition," *ACM Computing Surveys*, Vol. 17, No. 1 pp. 75-145, 1985.

[2] Will P.M., Pennington K.S., "Grid Coding: A Novel Technique for Image Processing," *Proc. IEEE*, vol. 60, no. 6, pp.669-680, 1972.

[3] Wang Y.F., Mitiche A., Aggarwal J.K., "Computation of Surface Orientation and Structure of Objects Using Grid Coding," *IEEE Trans. Pattern Anal. Machine Intell*, vol. PAMI-9, no. 1, pp.129-137, 1987.

[4] Posdamer J.L., Altschuler M.D., "Surface Measurement by Space-encoded Projected Beam Systems," *Comput. Graphics Image Processing*, vol. 18, pp.1-17, 1982.

[5] Boyer K.L., Kak A.C., "Color-Encoded Structured Light for Rapid Active Ranging," *IEEE Trans. Pattern Anal. Machine Intell*, vol. PAMI-9, no. 1, pp.14-28, 1987.

[6] Hu G., Stockman G., "3-D Surface Solution Using Structured Light and Constraint Propagation," *IEEE Trans. Pattern Anal. Mach. Intell*, vol. 11, no. 4, pp.390-402, 1989.

[7] Vuylsteke P., Oosterlink A., "Range Image Acquisition with a Single Binary-Encoded Light Pattern," *IEEE Trans. Pattern Anal. Machine Intell*, vol. 12, no. 2, pp.148-164, 1990.

[8] MacWilliams F.J., and Sloane N.J.A., "Pseudo-Random Sequences and Arrays", *Proc. IEEE*, vol. 64, no. 12, pp. 1715-1729, 1976.

[9] Petriu E., McMath W.S., Yeung S.K., Trif N., Bieseman T., "Two-Dimensional Position Recovery for a Free-Ranging Automated Guided Vehicle," *IEEE Trans. Instrum. Meas.*, Vol. 42, No. 3, pp.701-706, 1993.

[10] Hall E.L., Tio J.B.K., McPherson C.A., Sadjadi F.A., "Measuring Curved Surfaces for Robot Vision," *IEEE Computer*, vol. 15, no. 12, pp. 42-54, 1982.

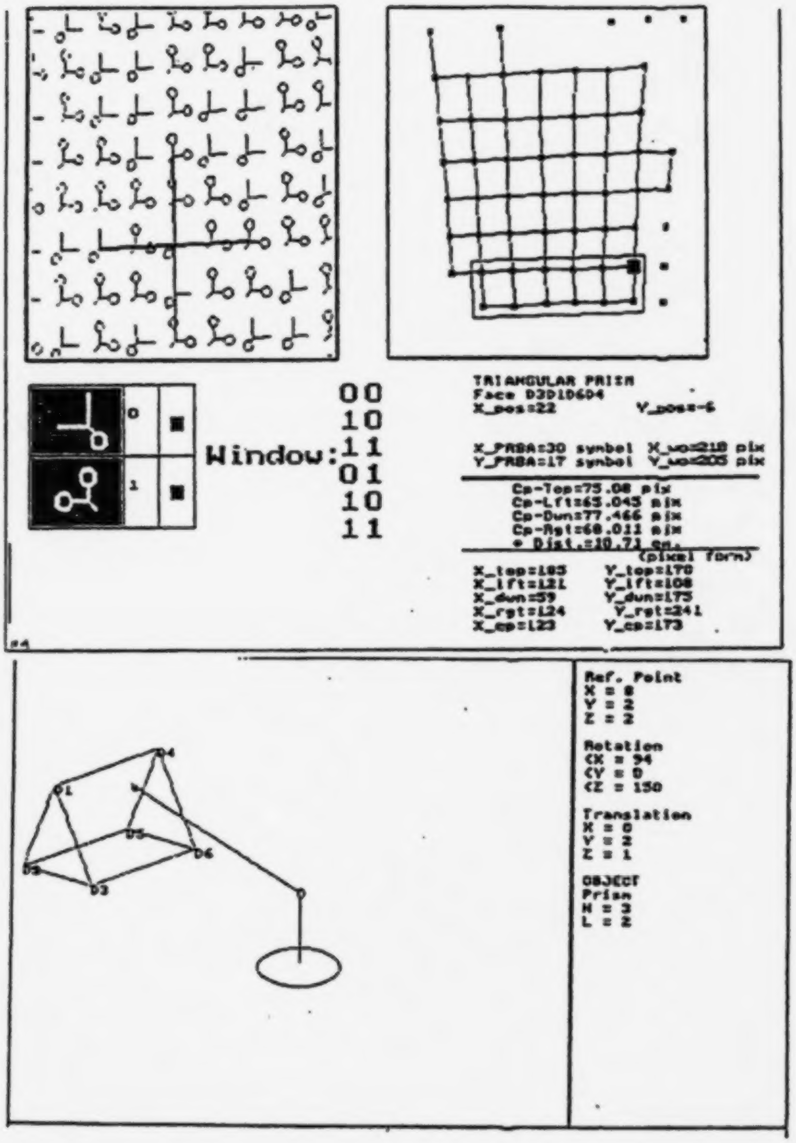
[11] Holt R.J., Netravali A.N., "Camera Calibration Problem: Some New Results," *CVGIP: Image Understanding*, vol. 54, no. 3, pp.368-383, 1991.

[12] Fischler M. A., Bolles R. C., "Random Sample Consensus: A paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography," *Comm. ACM*, Vol. 24, No. 6, pp. 381-395, June 1981.

[13] Ulupinar F. , Nevatia R. , "Perception of 3-D Surfaces from 2-D Contours," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 15, No. 1, pp. 3-18, January 1993.

OBJECT		VERTICES	
LABEL	TYPE	GEOMETRIC MODEL LOCATION (Fig.6)	P.R.B.A. (i,j) COORDINATES
PrismH3L2	Triangular Prism	D2-D1-D4	(00, 24)
		D1-D4-D5	(00, 32)
		D4-D5-D2	(15, 32)
		D5-D2-D1	(15, 24)
		D3-D2-D5	(00, 32)
		D2-D5-D6	(00, 40)
		D5-D6-D3	(15, 40)
		D6-D3-D2	(15, 32)
		D3-D1-D4	(11, 24)
		D1-D4-D6	(11, 32)
		D4-D6-D3	(27, 32)
		D6-D3-D1	(27, 24)

Fig. 7: Vertex mapping for the triangular prism named "PrismH3L2."



```
00
10
Window:11
01
10
11
```

```
TRIANGULAR PRISM
Face D3D1D6D4
X_pos=22            Y_pos=-6

X_PRBA=30 symbol  X_wo=218 pix
Y_PRBA=17 symbol  Y_wo=205 pix
      Cp-Top=75.08 pix
      Cp-Lft=63.045 pix
      Cp-Dwn=77.466 pix
      Cp-Rgt=68.011 pix
      * Dist.=10.71 cm.
                     (pixel form)
X_top=185      Y_top=170
X_lft=121      Y_lft=108
X_dwn=59       Y_dwn=175
X_rgt=124       Y_rgt=241
X_cp=123       Y_cp=173
```

```
Ref. Point
X = 8
Y = 2
Z = 2

Rotation
<X = 94
<Y = 0
<Z = 150

Translation
X = 0
Y = 2
Z = 1

OBJECT
Prism
N = 3
L = 2
```

Fig. 8: Visual recognition of the PRBA encoded triangular prism (experimental results).

90C 213

DND KNOWLEDGE-BASED SYSTEMS/ROBOTICS WORKSHOP

Projets belges MDN de développement en robotique intérieure et extérieure: des premiers pas de robots multipodes au contrôle intelligent de robots mobiles.

Baudoin, Yvan (*)
Ecole Royale Militaire
30 Av de la Renaissance
B 1040 Brussels , Belgique

#149694

(*) Directeur du Laboratoire de Dynamique des Structures et Robotique du Département de Mécanique Appliquée de l'Ecole Royale Militaire (Polytechnique) de Belgique.

Synthèse

L'Etat-Major Général des Forces Armées belges a autorisé le démarrage d'une étude consacrée à la robotique: cette étude étalée de 1993 à 1998 comprendra essentiellement trois phases: 1993 et 1994 seront réservées à la mise au point de robots statiques de laboratoire et de robots multipodes de dimension modeste, de commande électropneumatique et électrohydraulique, munis d'un maximum de senseurs de type différent, permettant: l'examen cinématique et dynamique de la mobilité, le contrôle extéroceptif des mouvements dans un environnement statique ou dynamique, incluant la génération de trajectoires, la mise au point enfin du traitement de signaux US (ultrasons), IR (infrarouges), CCD en vue de modéliser l'environnement. De 1995 à 1998, à partir des plateformes développées ou à partir d'une plateforme acquise dans le cadre d'une éventuelle coopération internationale fondée sur les recommandations du Panel VIII/RSG 18 OTAN, par exemple, l'expérience acquise sera "transférée" en environnement tout-terrain. En cette fin 1993, notre communication se limitera aux premiers résultats obtenus: la description d'hexapodes et de leur contrôle, la description des modèles environnementaux fondés sur l'acquisition ultrasonique, la description enfin de méthodes de générations de trajectoires en environnements faiblement dynamiques intérieurs.

Introduction.

Le développement extraordinaire de la microinformatique, de sa capacité et de sa rapidité, incite aujourd'hui de nombreux secteurs à investir dans la réalisation de machines automatisées dotées d'une intelligence artificielle toujours plus performante, renforçant le caractère autonome de celles-ci. Parmi ces machines, les robots mobiles autonomes (RMA) et, plus particulièrement, les robots multipodes suscitent un intérêt dicté par trois motifs au moins: (i) leur étude renforce notre connaissance du contrôle de la locomotion animale, ce qui intéresse directement la recherche biologique et, notamment, si on s'intéresse au point de vue de la Défense Nationale, le développement d'une chirurgie de réparation des traumatismes physiques consécutifs au combat, (ii) la mobilité de multipodes sur terrain accidenté, dans un environnement chaotique d'obstacles mobiles, est amélioré par une morphologie qui ne nécessite pas un contact continu des supports, (iii) enfin, et cet argument pour être philosophique n'en est pas moins pertinent, l'évolution des espèces n'a probablement pas favorisé par hasard ce type de mobilité. Six aspects techniques doivent être étudiés dans le cadre de la réalisation de robots mobiles, multipodes ou non: (i) le contrôle de la cinématique liée aux performances mécanique espérées (ii) la mise au point d'algorithmes de contrôle des mouvements nécessaires à une tâche donnée (iii) l'étude des capteurs proprio- et extéroceptifs de la machine (iv) la mise en forme et l'adaptation continue de la connaissance dont doit disposer le cerveau artificiel du robot (v) la programmation et le type de programmation ou de langages permettant la description de la mission confiée au robot et la décomposition de celle-ci en tâches élémentaires (vi) enfin l'intégration harmonieuse des solutions apportées aux cinq défis ainsi énumérés. Nous nous proposons d'en souligner quelques caractéristiques à partir des prototypes multipodes que nous mettons en oeuvre en ce moment.

Conception, cinématique et stabilité d'hexapodes.

C'est au milieu du 19ème siècle que les premiers mécanismes de transfert -déplacement de pattes- et transport -déplacement du robot sur pattes- furent introduits par Chebyshev, mais ce n'est qu'en 1970 qu'apparaît le contrôle informatique de la démarche d'un multipode, en l'occurence l'hexapode développé par McGhee [1]. Depuis, les études se concentrent sur les modes de déplacement (marche, saut, course) et le comportement qui en découle: lorsque le robot marche, les effets inertiels peuvent être négligés et l'étude peut se limiter à la cinématique et à la stabilité. Notre choix s'est d'abord porté sur la réalisation d'un hexapode d'architecture hexagonale, mu par actuateurs électriques gérés par microcontroleurs et commandés en 'pulse width modulation' (PWM): le prototype, dont le schéma et l'espace de travail en coupe verticale apparaît sur la figure 1 est actuellement mis en oeuvre dans le laboratoire de constructions mécanique et robotique (CMR) de l'Université Libre de Bruxelles (ULB) sous la direction du Professeur Preumont. Simultanément, un hexapode mu par actuateurs pneumatiques, de rapport coût-efficacité nettement plus avantageux (moins de 3.000 $US), était mis en oeuvre dans notre laboratoire (RMA): l'espace de travail y associé ne minimise pas aussi efficacement que dans le premier cas le couplage cinématique entre le mouvement horizontal et le mouvement vertical: ce découplage est pourtant avantageux dans la mesure où il minimise la consommation énergétique et simplifie la génération de trajectoires rectilignes horizontales, par action sur les degrés de liberté q2 et q3, l'articulation q1 restant bloquée pendant toute la phase de support du robot. Tous les points de l'espace de travail d'une patte peuvent être tabulés pour une série discrète du paramètre q1, en sorte que l'élongation radiale d'une patte puisse s'exprimer par $r_i = f(q_2)$ et sa hauteur $Z_i = f(q_1)$; Le déplacement du robot dans une direction déterminée se basera alors sur la résolution du problème cinématique inverse permettant le choix de q2 et de q3 pour toute valeur imposée de $X_i(t)$ et $Y_i(t)$. Le choix d'une démarche dans une direction donnée et la transition d'un type de démarche à l'autre nécessite de plus que soient étudiés les conditions de stabilité sur terrain plat et sur terrain accidenté.

Démarche en terrain plat. [2]

Lors de la marche, la durée d'un cycle T comprenant le temps de support s et celui de transfert t leur est liée par la fraction active de la patte ou facteur de service b selon:

$$s = b.T \quad \text{et} \quad t = (1-b).T \qquad (1)$$

Si R désigne la course d'une patte, la distance parcourue durant un cycle vaudra alors:

$$L = R/b \qquad (2)$$

La vitesse du robot, limitée par les actionneurs (typiquement 0,1 m/s), s'exprimera alors par:

$$V = L/T = R.T\,/\,b = (R/t).\,(1-b/b) \qquad (3)$$

Cette formule, applicable aux hexapodes pour une valeur du facteur de service supérieure ou égale à 1/2, montre que la vitesse est indépendante du séquencement du mouvement des pattes. Par contre, la stabilité statique assurée par un minimum de trois pattes en contact avec le sol, sera tributaire du séquencement. Ce séquencement est généralement identifié par le déphasage Øi des pattes i par rapport à la première. En adoptant la numérotation de la figure 2, un séquencement régulier (même facteur de service par patte), symétrique (déphasage d'un demi-cycle entre pattes gauches et droites de même rang 1/6, 2/5 et 3/4) ,caractérisé par b (exprimé par s/T en 1/12) peut être représenté par le diagramme de la figure 3. La connaissance de Ø2 et Ø3 définira le type de démarche et la position du polygone de sustentation dont la position varie brutalement lorsqu'une patte se pose ou se lève. Il importe naturellement que la projection du centre de gravité reste intérieure à ce polygone: la stabilité statique est, de ce fait, souvent mesurée [3] par la marge de stabilité frontale réduite définie par la valeur minimale, au cours du cycle, de la distance entre la projection du centre de gravité et le polygone, dans le sens du mouvement (Sl, FMS sur la figure 2), rapportée à la distance parcourue L :

$$S^* = Sl/L \qquad (4)$$

dont le minimum est toujours obtenu avant la pose d'une patte. De l'examen géométrique d'une configuration particulière, comme celle de la figure 4, avant la pose de la patte 1 tandis que la patte 2 est sur le sol, et de l'exploitation des formules de base (1) à (4), on déduit, pour un déphasage Ø2 > 1/2 et un facteur de service b > Ø2 :

- la distance déjà parcourue par la patte 6 : (R/bT).(T/2) = R/2b
- la distance parcourue par la patte 2 : Ø2 en retard ou (1-Ø2) en avance : R.(1-Ø2)/b
- la marge de stabilité critique correspondante:

$$\begin{aligned} Sl &= 1/2\,.(R/2 - R(1-\varnothing 2)/b \\ &\qquad + P - R/2 - Rb/2) \\ \text{ou} \quad S^* &= b.Sl/R \\ &= P.b/2R + b/2 - 3/4 + \varnothing 2/2 \qquad (5) \end{aligned}$$

Ce type de raisonnement peut s'appliquer aux différentes démarches réalistes, parmi lesquelles les démarches d'onde vers l'avant (forward wave gait ou FWG) telles qu'illustrée par la figure 3 (Ø2 = b > 1/2 et Ø3 = 2b-1), d'onde vers l'arrière (Ø2 < 1/2: la pose successive des pattes d'un côté apparaît comme se propager de l'avant vers l'arrière), à incrément de phase constant (Ø3-Ø2 = Ø2-Ø1). Notons enfin le cas particulier de b=1/2 pour lequel deux triplets de pattes alternent en phase de transfert et de support: cette démarche adoptée pour l'hexapode pneumatique porte le nom de tripode alterné.

Démarche en terrain accidenté [4]

Outre la stabilité examinée ci-dessus, et dans l'hypothèse où l'hexapode ne dévie pas de la trajectoire (en ligne droite) qui lui est imposée (angle de lacet , yaw y, nul) , le contrôle de l'attitude requiert la détermination de la coordonnée Zg du centre de gravité et celle des angles de roulis (roll r) et de tangage (pitch p). Cette détermination se fonde actuellement sur le choix du trépied minimal (3 pattes au sol définissant un polygone contenant la projection du centre de gravité) qui d'un instant t à l'instant suivant t+dt (exprimé en 1/12 de temps de cycle) minimalise la variation RMS (root mean square) d'attitude ou:

$$\sqrt{\{p(t+dt)-p(t)\}^2 + \{r(t+dt)-r(t)\}^2}$$

Sur un sol numérisé par éléments finis et par rapport à des systèmes d'axes absolu et relatifs liés aux trois pattes posées sur le sol (figure 5), il est alors simple de décrire un système de 9 équations non linéaires à 9 inconnues (coordonnées des 3 pattes, altitude du centre de gravité, roulis et tangage) résolvable par la méthode classique de Newton-Raphson. La matrice Jacobienne contient des dérivées partielles de matrices de rotation par rapport aux angles de roulis ou tangage dont le calcul apparemment complexe peut être évité en constatant que ces dérivées correspondent aux mêmes matrices où l'angle par rapport auquel on dérive est incrémenté de 90°. Ainsi, si R définit la matrice de rotation du référentiel 2 par rapport au référentiel 1:

$$R = R(r,p,y) = R(y).R(p).R(r)$$
$$\longrightarrow dR/dp = R(p+\pi/2)$$

Au stade actuel de conception et de réalisation en laboratoire, compte tenu des considérations résumées ci-dessus, un hexapode à motorisation électrique commandée à partir de microcontroleurs embarqués fonctionne. Six types de démarche stable sur sol plat ou légèrement accidenté peuvent être choisis: la transition optimale entre types de démarche sur sol plat ou quelconque n'est pas encore résolue. A cet effet, un programme de simulation est développé, incluant l'aspect réactionnel des pattes en contact avec le sol. Un hexapode électropneumatique fonctionne également, sous le contrôle d'un automate programmable. Différents types de démarche caractérisés par un facteur de service égal à 1/2 sont disponibles, sur sol plat. Cet hexapode est actuellement équipé d'une caméra automatiquement orientée dans l'une des six directions (six pattes) que peut suivre le robot. Le pilotage de ce robot est assuré par joystick. Enfin, un prototype hexapode d'architecture rectangulaire, équipé d'actuateurs hydrauliques et d'une centrale de puissance entraînée par moteur à combustion interne (MCI) ,et destinée à se mouvoir en environnement extérieur, est pratiquement réalisée. Cette plateforme ne sera mise en oeuvre qu'à la mi-94, après développement d'un minimum d'algorithmes et acquisition d'une expérience de laboratoire. Il va de soi que les principes définis ci-avant valent pour la locomotion de cette plateforme (moyennant numérotation cohérente des pattes).

Notre préférence pour le développement de plateformes multipodes s'explique surtout par le fait que la commande de roues, de chenilles ou de combinaisons de ces deux moyens classiques est quasi universellement connue: de nombreux châssis existants peuvent d'ailleurs être adaptés. De plus, les robots multipodes peu coûteux peuvent être "sacrifiés" sur sites dangereux, parfois difficilement accessibles. (sites nucléaires accidentés ou zones urbaines minées).

Perception de l'environnement, stratégies de mouvement.

La figure 6 résume le système de navigation modeste que nos premières activités en laboratoire ont déjà permis de développer. Notre but final est de construire un logiciel de navigation guidant un robot mobile autonome dans un environnement inconnu à priori, peuplé d'obstacles mobiles. La première philosophie retenue est généralement désignée de "bottom-up approach" [5] et construit séquentiellement une représentation de l'environnement et une planification du mouvement.

Le robot de laboratoire est équipé d'un capteur à ultrasons Polaroïd monté sur un servomoteur à courant continu et balayant l'environnement en 18 secteurs de 20°. Le cône d'ouverture du capteur, caractérisé par une dispersion d'énergie donnée par le diagramme d'émission classique d'un tel capteur, introduit, dans une modélisation sous forme de grille et conforme à l'approche définie par Elfes [6], un recouvrement entre cellules de secteurs contigus d'une part et une incertitude sur l'occupation réelle d'une cellule par un obstacle d'autre part.

Notre modèle le plus simple attribue à l'état d'une cellule (50 x 50 cellules de 10 cm x 10 cm) un coefficient d'occupation p-out(i,t) (i=1,2500) variant de 0 (cellule certainement libre) à 100 (cellule certainement occupée), la valeur 50 identifiant une cellule inconnue. Ces valeurs extrêmes ne valent que dans l'axe de visée tandis qu'une décroissance linéaire est appliquée en fonction de l'écartement de la "droite discrétisée de cellules" par rapport à l'axe de perception. Il en résultera une distribution de confiance (c(i)=Abs(50-p(i)) calquée sur la distribution d'énergie du signal émis. Au cours du déplacement du robot, comme d'ailleurs lors de chaque perception dans les zones intersectorielles contigües, la grille doit évoluer en vertu de règles logiques; nous les avons aussi voulues simples, illustrées par la figure 6, car elles combinent les coefficients d'occupation par:

$$p\text{-}out(i,t+1) = p\text{-}out(i,t) + (p\text{-}in(i,t+1) - 50) \qquad (6)$$

(< 100 ou max = 100 si >0
ou min =0 si <0
avec p-in : nouvelle perception)

Lors du déplacement du robot, la carte de l'environnement (t) est translatée du pas accompli, la grille (t+1) ensuite calculée: pour accélérer les calculs, on peut aussi limiter le calcul à une zone perçue réduite dans la direction suivie par le robot, et, dans ce cas adapter quand même suivant cette règle (avec p-in(i,t+1) = 50) les cellules non "perçues" .

Stratégie locale du mouvement.

Après chaque mouvement, de nouvelles informations sont donc acquises et intégrées dans la carte de l'environnement, suivant la règle simple (6) dont un nouveau modèle, de type neuronal [8], est à l'étude. Le calcul d'une trajectoire est donc vain: il suffit, pour la configuration courante du robot, dans un environnement dynamique, de calculer la direction la plus prometteuse. Parmi les méthodes classiques décrites dans d'excellents ouvrages comme [7], par exemple, la méthode du potentiel ne nécessite pas l'acquisition de la connectivité de l'espace libre: le robot passe d'une configuration à la suivante en basant son choix sur le calcul du gradient de potentiel, dépendant uniquement de l'état de l'environnement dans le voisinage. Il importe cependant de minimaliser la durée du calcul. A cet effet, l'information donnée par la carte de l'environnement ne sera pas consultée en entier. Un "disque" de cellules (12 cellules de rayon), centré sur le robot, appelé la zone d'action du potentiel, est défini. Comme le montre la figure 7, cette zone d'action (en grisé) est élargie à un carré de 25x25 cellules pour simplifier le mode d'adressage des cellules. Les cellules situées au delà du rayon d'action seront éliminées par la suite. La zone d'actions est partagée en secteurs d'ouvertures égales, centrées sur le robot, en nombre égal au nombre de perceptions constituant un profil complet, soit 18. On attribue à chaque secteur un potentiel répulsif en additionnant les potentiels des cellules occupées comprises dans ce secteur. Le potentiel d'une cellule étant, lui, une fonction du degré de confiance (ci) lui associé. Les 18 valeurs résultantes du potentiel constituent l'histogramme de répulsion (figure 8) Soient p-out(i,t) et c(i,t) respectivement le coefficient d'occupation et le degré de confiance associé de la cellule i et E = { i : p-out(i,t) > 50 } l'ensemble des cellules probablement occupées dans la zone d'action, la valeur du potentiel répulsif associé à toute cellule i de E sera donnée par:

$$\begin{aligned} rep(i,t) &= r(i).(p\text{-}out(i,t) - 50) \\ &= r(i).c(i,t) \qquad (7) \end{aligned}$$

où r(i), valeur du champ répulsif marginal, est défini pour un degré de confiance unitaire (c(i,t)=1) et lu dans une table mémorisée. Cette table est en fait un calque qu'on superpose à la zone d'action. Un sous-secteur de potentiel marginal égal à 20 possèdant un rayon double de celui de la projection supposée circulaire du robot assure l'évitement des obstacles proches du robot. Au delà de ce rayon, le potentiel marginal décroît proportionnellement au carré de la distance à la cellule centrale. Si, maintenant, Sk désigne le secteur k (k=0,17 voir fig 8) de la zone d'action, le programme "superpose" la table des secteurs à celle des potentiels marginaux et calcule (très rapidement):

$$rep\text{-}k\,(t) = \sum_{i \in Sk} rep(i,t) \qquad (8)$$

L'histogramme obtenu ou vecteur à 18 composantes est ensuite lissé plusieurs fois selon une moyenne mobile:

$$\begin{aligned} rep\text{-}k'\,(t) = 1/3.\ (&rep\text{-}k\text{-}1(t) + rep\text{-}k(t) \\ &+ rep\text{-}k\text{+}1(t)) \end{aligned} \qquad (9)$$

qui assure une compensation des erreurs de discrétisation d'une part et, surtout et d'autre part, un tassement de la courbe permettant d'éviter le mouvement dans une direction trop proche de l'obstacle (figure 9.a obtenue en limitant le nombre de lissages à 4 tandis que 9.b correspond à 6 lissages) La figure 10 exprime aussi visuellement l'heuristique de choix du cap minimum privilégiant dans l'ordre la direction du "but" à atteindre, la direction d'angle minimal entre le cap précédent et le dernier secteur autorisé, enfin l'angle minimal entre direction du but et directions multiples offertes par l'histogramme. La stratégie locale garantit un guidage efficace dans un champ d'action restreint: un planificateur global, poseur de jalons, càd objectifs intermédiaires, et garant du contrôle effectif du robot est nécessaire. Nous avons, pour définir celui-ci, adopté un graphe de rétraction obtenu par segmentation de la carte de l'environnement. On extrait à partir de la carte un "squelette" obtenu par intersection d'ondes qui se propagent à partir des obstacles ou cellules occupées (p-out(i,t) >50): cette propagation consiste en l'attribution d'une "Manhattan distance" ou distance discrète égale à la somme des écarts entre les coordonnées correspondantes des cellules entre lesquelles elle est calculée. La figure 11 illustre graphiquement le calcul à deux instants donnés. Sur le squelette obtenu (quasi équidistant des zones occupées) dont les cellules sont supposées correspondre au centre de gravité du robot (et qui ne sont donc retenues dans le squelette que si leur distance aux cellules-obstacles sont suffisantes), le planificateur peut à présent définir des jalons considérés comme buts pour le planificateur local

Conclusions et perspectives.

Si l'approche "bottom-up" a d'abord été privilégiée au cours de nos premiers travaux, elle ne le restera probablement pas longtemps car la séquence "perception - modélisation - plannification -perception - ..." ne répond pas réellement au mode naturel de réaction intelligente: en fait, la mémoire du robot doit déjà disposer d'informations types (apprises) sur l'environnement et les comportements à développer dans cet environnement (trajectoires types y associées, par exemple). La phase de perception doit alors permettre un calcul rapide de proximité (ou d'identité ou de similitude) entre la carte enregistrée et les cartes mémorisées. Dans le cas d'une grille de 2500 cellules, par exemple, ce calcul pourrait se borner à la somme des carrés des différences entre p-out(i,t) et les valeurs 0 ou 100 de cellules de n cartes mémorisées relatives à l'application donnée. Ainsi, sur base de telles idées, notre recherche actuelle tente donc l'approche dite "top-down".

Outre les tendances de recherche définies ci-dessus, trois laboratoires de l'Ecole Royale Militaire développent leur recherche dans les domaines du traitement d'images (laboratoire du Professeur Acheroy), du traitement de la parole (laboratoire du Professeur Vloeberghs) et du développement de systèmes-experts (laboratoire d'intelligence artificielle du Dr Ir Leysen): ces recherches parallèles ne convergeront vers la robotique mobile qu'après la mise au point d'une plateforme mobile équipée de senseurs multiples et capables de mouvements en environnement extérieur, soit vers la mi-94. Nul doute que les contributions des trois laboratoires cités seront présentés lors du prochain "workshop".

Remerciements.

L'auteur remercie particulièrement le Professeur Dr Ir Preumont de l'Université Libre de Bruxelles pour l'étude particulière de la stabilité des hexapodes, ainsi que Messieurs Ghuis, Hallbach, Malékian et Bosschmann leurs assistants pour la programmation (turbo C) des logiciels impliqués par l'exposé précédent.

Bibliographie.

[1] Mc Ghee -"R.B. Vehicular legged locomotion" - Adv.Automation and Robotics - 1983

[2] A.Preumont, D.Ghuis, Y.Baudoin "Etude cinématique et de stabilité des hexapodes" Rvue of Mechanical Engineers 2 Sep 1991.

[3] A.Preumont, P.Alexandre,D.Ghuys "Gait analysis and implementation of a six leg walking machine" Int.Conf.on Advanced Robotics, Pisa, June 92

[4] Ch.Malékian "Simulateur robot marcheur hexapode" Rapport interne ERM/ULB Aout 92 (disponible sur demande)

[5] James S.Albus "Robotics, past, present and future" Agard LS 142 Oct 1985

[6] A.Elfes "A sonar based mapping and navigation system" 1986 IEEE Int.Conf.on Robotics, San Francisco

[7] J.C.Latombe "Robot motion planning" Kluwer Academic Publishers ISBN 0-7923-9129-2

[8] Y.Baudoin, A.Preumont "From electro-pneumatical to electrical six-legged robot : optimalisation of the sensor-based proprioceptive (kinematical) and exteroceptive (neurological) control" SPIE/Mobile Robots VIII, Boston, Sept 1993.

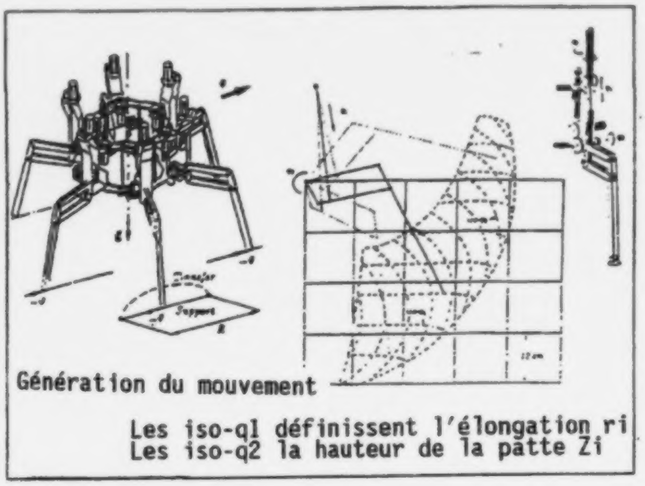

Transfer
Support
12 cm
Génération du mouvement
Les iso-q1 définissent l'élongation ri
Les iso-q2 la hauteur de la patte Zi

Fig 1. Hexapode et espace de travail

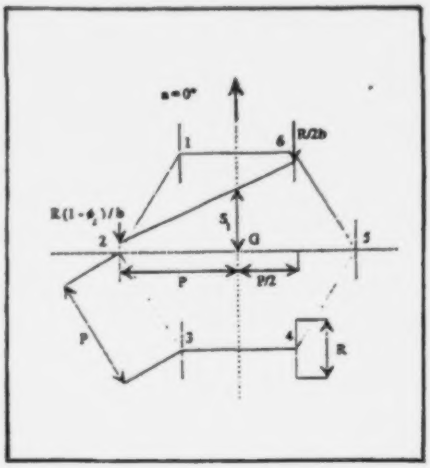

α=0°
1
6
R/2b
R(1-φ)/b
2
S
G
5
P
P/2
P
3
4
R

Fig 4. Examen de la stabilité

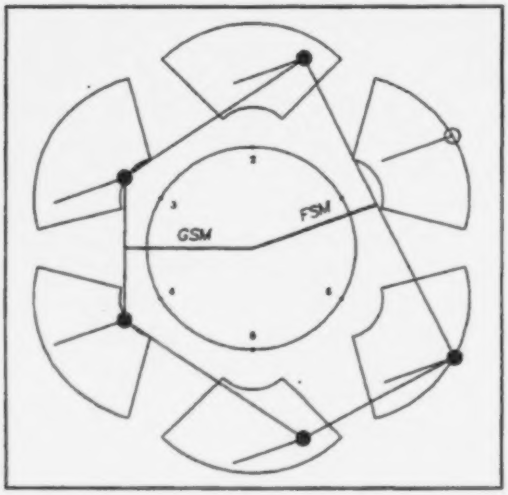

2
3
GSM
FSM
4
6
5

Fig 2. Numérotation (FSM: marge de stabilité frontale)

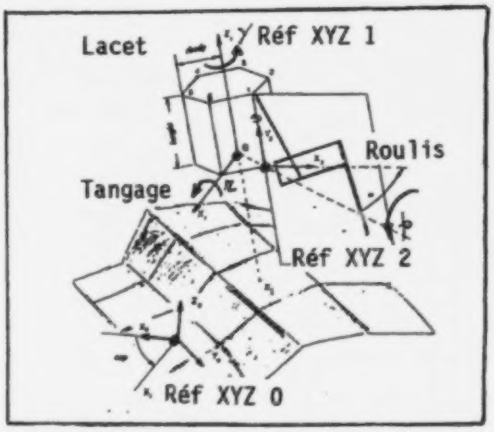

Lacet
Réf XYZ 1
Roulis
Tangage
Réf XYZ 2
Réf XYZ 0

Fig 5. Démarche sur sol accidenté

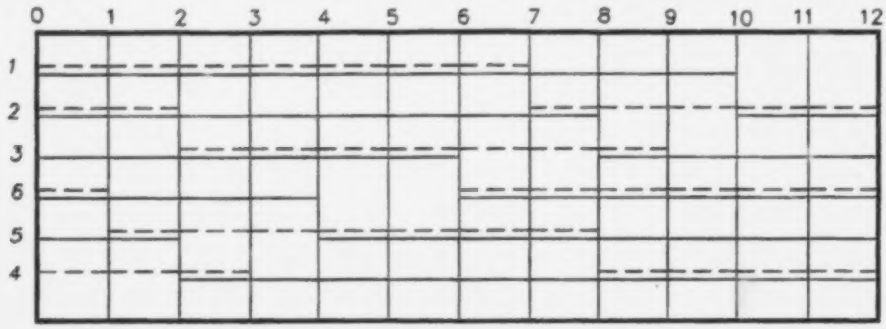


Fig 3.Diagramme de séquencement d'une démarche

———— : FWG 10/12 ------ : Eq.Ph.G 7/12

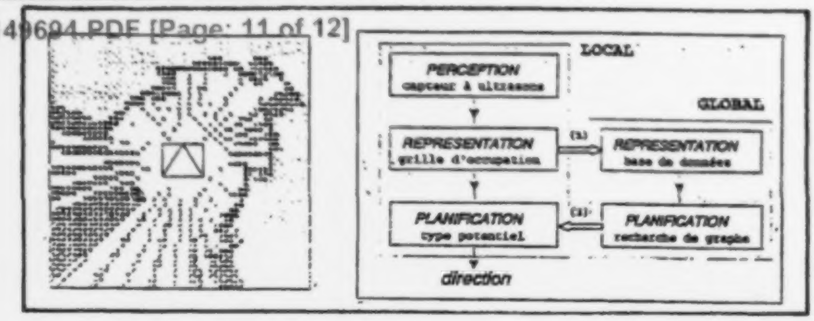

LOCAL
PERCEPTION
capteur à ultrasons
GLOBAL
REPRESENTATION
grille d'occupation
(1)
REPRESENTATION
base de données
PLANIFICATION
type potentiel
(2)
PLANIFICATION
recherche de graphe
direction

Fig 6. Grille d'occupation

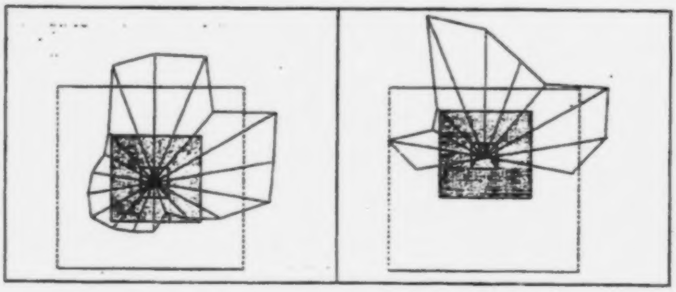


Fig 7. Zones de calcul de répulsion.

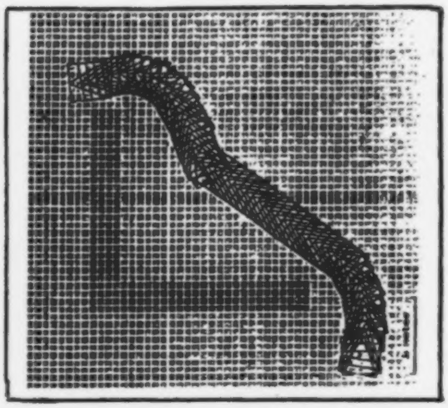


Fig 9.b Trajet à 6 lissages

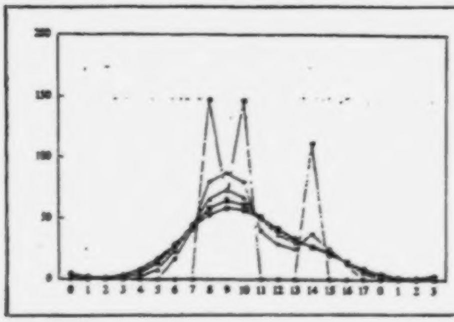


Fig 8. Histogramme de répulsion

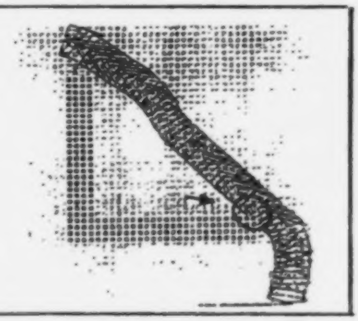


Fig 9.a Trajet à 4 lissages

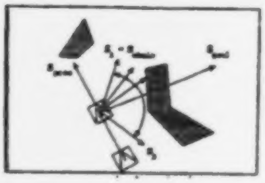


Fig 10. Heuristique

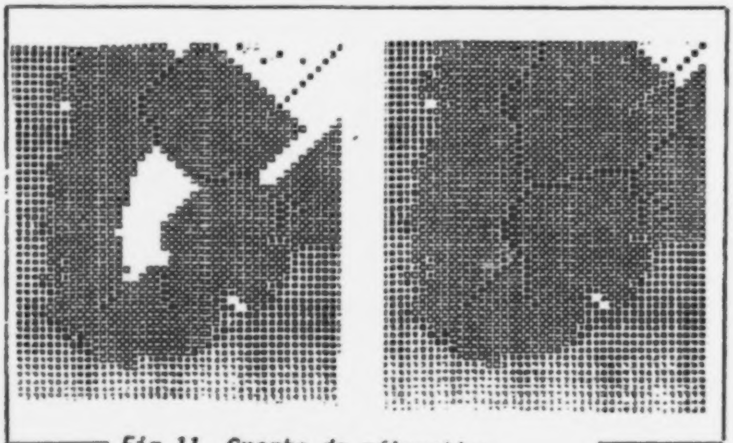


Fig 11. Graphe de rétraction

906986

DND Knowledge-Based Systems and Robotics Workshop, November 1993

NATO Research Study Group on Operator-Robot Interaction

Julius J. Grodski
Defence and Civil Institute of Environmental Medicine
PO Box 2000, Downsview, Ontario, Canada, M3M 3B9
email: jul@dciem.dnd.ca

#149695

Abstract

Panel 8 on the Defence Applications of Human & BioMedical Sciences, a part of the NATO Defence Research Group proposed a Research Study Group (RSG) on telerobotic systems in 1988. Terms of Reference (TOR) for the group were developed by an Exploratory Group. Upon approval of the TOR by Panel 8, the RSG-18 on Operator-Robot Interaction (ORI) was formed with membership from Canada, France, Germany, the Netherlands, U.K., US Air Force, US Army, and US Navy. Belgium joined later and US Navy dropped off. The RSG-18 held seven semi-yearly meetings to consider ORI issues affecting effectiveness of telerobotic systems. A workshop involving invited experts was organized and held to obtain additional input to the topic. The main operational applications were identified by a subgroup of military users. The other subgroups considered function allocation between man and robotic subsystems, driving and emplacing a robotic platform using a high bandwidth video/control link, driving and emplacing a remote platform using a low bandwidth video/control link, and mission module man-machine interface. Some details of the workshop, its results, as well as conclusions and recommendations resulting from the RSG-18 are presented. A complete report on the work of the RSG-18 will be published as a NATO DRG report.

Introduction

The NATO Defence Research Group is responsible for promoting co-operation in research and technology which might lead, in the long term, to future equipment. One of the main objectives of the Group is to review and advise on the possible military consequences of advances in the fields of science and technology. The Group can undertake research studies on its own initiative or at the request of other NATO Groups and can identify and promote suitable areas for bilateral or multilateral co-operation in defence research and technology. The Group is also the only NATO body which sponsors Research Study groups (RSGs).

The oversight of specific topics, however, has been delegated to Panels as subordinate bodies. There are a number of Panels, most of them are fairly specific either to a field of science, e.g. optics or to applications area, e.g. air defence. Panel 8 on Human and BioMedical Sciences has probably the widest scope.

Exploratory Group (EG)

In 1986, Panel 8 received a proposal from Germany for a new activity on "Soldier-Robot Interaction". The Panel agreed that the topic was interesting and invited Germany to prepare a more detailed proposal and the interested nations to designate points-of-contact.

The next year Panel 8 established an Exploratory Group (EG) for the theme. Germany was the pilot nation for the EG. Canada, France, Netherlands, United Kingdom, and the United States joined the EG. Panel 8 directed the EG to produce a program of work, narrative rationale and the terms of reference (TOR) for an RSG titled Operator-Robot Interaction (ORI).

The EG reported to Panel 8 late in 1988 that an initial exchange of information took place among the interested nations and that further cooperation was desirable since many nations were considering the development of telerobotic systems (TRS). The EG recommended that an RSG be established to identify and prioritize critical unresolved ORI issues.

Consideration of TRS for military applications took place against the background of the following assumptions. TRS operating in unstructured environments will not be fully autonomous for quite some time. They will depend upon a close cooperation with the human operator who will define, control and monitor missions, and select the appropriate degrees of autonomy for each mission. The operator will decide when to start the mission and judge whether it was a success.

The potential military advantages of such systems were considered to include:

(a) Increasing operator effectiveness and survivability by removing operators from increasingly lethal or hazardous environments;

(b) Multiplying effectiveness with fewer personnel and the option of controlling multiple devices from a single control station.

(c) Providing long-term capabilities in sustained operations.

The TOR for the recommended RSG stipulated that the principal objective of the RSG was to establish an active exchange of information among member nations and to stimulate the military application of human-compatible telerobotic systems. The RSG was to identify the present state of knowledge, future trends, and unresolved problems area in TRS, and identify and analyze critical unresolved ORI issues.

Figure 1 shows a general schematic of a telerobotic system, including the interface between the human operator and the robotic system. The interface must be compatible with the overall requirements of system and therefore it needs to satisfy the functionality and capability requirements imposed or expected for the overall system.

The operator-robot interaction (ORI) issues are largely connected with the interfaces but are also strongly shaped by the overall system. They include the human factors matters of the design of displays, controls and the work environment. Other factors affecting the issues originate both from the human side (such as e.g. fatigue, physiological measures) and the robotic side (such as e.g., control allocation, world model).

Activities of the RSG

Panel 8 directed in 1989 that a new RSG-18 proceed with its work. The RSG membership was the same as that of the preceding EG. Belgium joined the RSG at its fifth meeting in May 1992.

Meetings of the RSG were held at semi-annual intervals from the spring of 1990 to 1993 and rotated between sites of the member nations. These meetings included visits to laboratories where robotics research and related activities take place. This aspect together with the program of presentations of national interests in robotics in general and specifically on ORI issues, provided an excellent opportunity for familiarization with the current work in the member nations.

The RSG drafted a program of work based on the direction that the primary objective was to identify unresolved ORI issues. In order to identify the applicable domain of the subject matter, some members of the RSG developed taxonomies of robotic systems in their nations and presented them to the RSG. Surveys of national ORI research were developed and presented by the members; the USA status quo was presented for each: Army, Air Force and Navy. Identified issues were assembled into a so-called "microstructure list" which addressed the modes and aspects of information feedback to operator, operator output, and system issues, such as e.g., latency latency, level of autonomy, etc. Thereby, it was

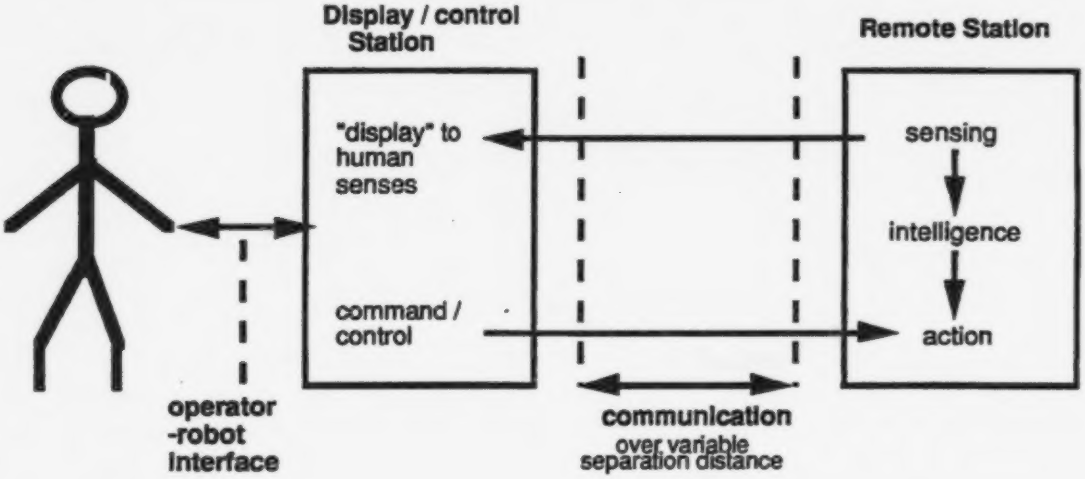

Display / control Station
Remote Station
"display" to human senses
command / control
sensing
intelligence
action
operator -robot interface
communication
over variable separation distance

Figure 1. Outline of a telerobotic system including the operator-robot interface (ORI).

was recognized that ORI issues involve not only the controls, displays, and workspaces but also matters of *functionality and capability* of the telerobotic control systems as they interact with the human operator and his capabilities and limitations. Core ORI issues also evolved from integration of issues in specific cases into generic themes, such as e.g.: information transfer (to the operator) or restricted sensing (of the remote platform), etc. It was observed that the core issues, however, are modulated by the mission type: placement of a TRS (which involves travel to an operation site) or execution of a task involving mission module control activities conducted at a remote operations site. Obviously, the core issues are intertwined with the full range of the ORI issues listed in the "microstructure list". A complete list of the core issues, the "microstructure list", etc., will appear in the NATO DRG report on the work of the RSG-18.

In order to identify and prioritize unresolved ORI issues for constructive international collaboration, it was considered essential to constrain the topic. It was therefore concluded that:

- a clear military perspective of TRS applications should be developed,
- a common outlook between the users of the participating members should be established, and
- on such a foundation, the outstanding ORI issues could be identified for the common outlook.

These objectives, however, waere impossible to achieve directly within the RSG for a few reasons; one of them was that RSGs do not normally involve military users. Also, RSGs have no budget of their own or other resources for resolution of such a complex issue. However, a practical method to achieve a resolution of the objective was available in form of a workshop. Within the RSG-18, the idea of a workshop was first discussed during the Toronto meeting in the fall of 1990. This concept evolved and a Workshop on Critical Issues in ORI was organized.

Workshop on Critical ORI Issues

France organized and hosted the Workshop on Critical ORI Issues in Bordeaux in October 1992. The Workshop involved experts from the operational user community, experts from the technical development community, and scientists from the community conducting applied research on human factors of "unmanned" systems. Attendance was limited to about 50 invited participants. The seven NATO nations with membership in the RSG-18 were represented at the Workshop.

Five subgroups (SG) were formed to conduct detailed discussions on specific technical-operational ORI issues. The SG themes were as follows:

- SG-1 operational validation
- SG-2 planning, navigation and function allocation
- SG-3 Vehicle MMI (High bandwidth)
- SG-4 Vehicle MMI (Low Bandwidth)
- SG-5 Mission Module MMI (High and Low Bandwidth)

(MMI= Man-Machine Interface)

Participants were assigned to one of the SG, prior to the Workshop, with the aim to:

- produce SGs with approximately equal membership, and
- match as closely as possible the expertise of the delegates with the theme of the assigned SG.

The Canadian representation was limited to four invitees. They included:

SG-1: LCol J.G. Lindsay, DLAEEM 4,
SG-2; Mr. R.H. Chesney, Defence Scientist, Defence Research Establ. Suffield,
SG-3: Prof. P. Milgram, Head, Ergonomics and Telerobotics Control (etc) Lab., University of Toronto;
SG-5: Dr. J.J. Grodski, Head, Telerobotics, DCIEM.

The Workshop involved opening and concluding plenary sessions and SG working sessions. All invitees presented short position papers to a full assembly at the opening. The Workshop considered ORI issues in all three areas of operational environment for application of unmanned vehicles: land - Unmanned Ground Vehicle (UGV), air - Unmanned Air Vehicle (UAV), and sea - Unmanned Underwater Vehicles (UUV).

The SGs were expected to produce the following outcomes:

- SG 1 was expected to suggest a list of priority mission scenarios and a strategy for robotic operations and optimal performance assessment .
- SG 2 was expected to identify current limitations on robotic deployment due to technology or inadequate prior data, and areas in which intermediate and high level ORI research is required.
- SG 3 was expected to conclude on the priority design features of a high bandwidth teleoperated driving station and to establish which of those elements require further research.
- SG-4 was expected to indicate whether low bandwidth teleoperation is a feasible operating mode, and in what circumstances it may be used. They should identify those areas of ORI research that need to be addressed for optimal use of low bandwidth information.

- SG-5 was expected to list the technology requirements for remote operations, list the current shortcomings, and recommend the thrusts for future technical and ORI research required to overcome those shortcomings in the next 3 to 5 years .

The SG1 identified and recommended to the other SGs that their considerations be based on three highest priority applications of unmanned systems:

- unmanned ground vehicle for reconnaissance, surveillance and target acquisition (RSTA) missions;
- unmanned aerial vehicle for RSTA missions;
- telerobotic mine/ordnance detection, emplacement and/or disposal.

The detailed conclusions of the SGs are too lengthy to quote and they will appear in time as a NATO publication. In general terms, however, it can be stated that the Workshop identified the highest priority ORI topics, and provided a range of primary conclusions and recommendations.

The major themes of the priority ORI topics include:

- simulation with man in the loop
- interoperability
- pre communication and pre display data processing
- mission and automation level relevant displays
- mission and automation level relevant control
- workstation design
- performance analysis methodologies.

Some of the more general findings generated by the Workshop include the following:

- Robotics is an emerging technology that offers many potential benefits to the NATO military user community. Key benefits are removing personnel from especially hazardous duties, and multiplying the effectiveness of dwindling military forces.

- Integration of robotics platforms, systems and mission modules into the military will require an iterative process of prototype development (strongly based on operational requirements), field testing, and incremental improvement, along with modification of existing battlefield tactics and doctrine to accommodate this new technology.

- Specifications, evaluation schemes and norms must be developed for military robotics systems. This is an area in which considerable opportunity exists for international cooperation and shared research.

- Commonality of controls and display configurations and design would go a long way toward helping the operators of the different systems to share information and perhaps mission tasks (both planning and execution).

- The role of the operator will change significantly from that of a remote, full time in-the-loop controller to that of a supervisor of one or more remote systems or platforms as robotics systems become increasingly autonomous. Research in the area of mental and physical workload assessment will be required to provide information critical to task analysis and allocation of functions between the operator and the robotics system, but the level of understanding and the associated models of mental workload are still at a relatively primitive stage. Much work needs to be done in this area.

Post-Workshop RSG Activity

The RSG-18 met after the Workshop. It assembled the Workshop results and considered the R&D implications of the Workshop. The RSG identified a number of potential research topics with the capability for multi-national involvement. Most of these topics are of potential interest to the Canadian R&D effort and they include: sensor data fusion, enhanced depth displays, video overlay displays, performance analysis methodologies, predictor displays, augmented reality displays, obstacle avoidance, mission module/sensor control. It is important that a majority of these themes are domain independent, i.e. they may have a potential applications in either land, air, or sea environments.

The life of the RSG-18 ended with its seventh meeting earlier this year. In view of the significant outstanding issues identified in this area and numerous common interest topics in this domain, RSG-18 is recommending to its parent body, Panel 8 that the work ought to be continued. Specific recommendation calls for establishment of a suitable RSG. There is also a high probability of even broader involvement of NATO members in such a pursuit.

Conclusion

The necessity to obtain value for money in defence procurement is a common goal for many NATO countries, including Canada. For an emergent topic, such as teleoperated systems, there is real potential to achieve this through the sharing of national expertise without the associated limitations of applicability that arise in more mature fields.

Canada has a significant potential to contribute to multinational efforts in this area and more importantly, it can significantly benefit from such engagements. It is therefore important that a continuity of the involvement is maintained and supported.

Acknowledgments

The work described here was carried out by the members of the NATO, Panel 8, RSG-18 under Chairmanship of Dr. D. Hodge of the US Army Human Engineering Laboratory and it includes efforts contributed by the participants of the Workshop on Critical ORI issues. The preceding work of the related Exploratory Group, chaired by Dr. K.-P. Holzhausen of Germany, is also acknowledged.

425747

DND Knowledge-Based Systems and Robotics Workshop, November 1993:

U.S.-Canada Bilateral Cooperation Opportunities in Defence Robotics

Russell M. Phelps
U.S. Army Research Laboratory
Aberdeen Proving Ground, Md. 21005-5066, USA

Julius J. Grodski
Defence and Civil Institute of Environmental Medicine
PO Box 2000, Downsview, Ontario, Canada, M3M 3B9

#149696

Abstract

Robotics is a key emerging technology where successsful implementation depends on the user's ability to effect advantageous solutions at the lowest possible development cost. This implies the need to the use the best and most suitable technologies, irrespective if their origin is national or foreign, thereby calling for effective cooperative international efforts. Robotics is a multidisciplinary domain with numerous subspecialties. This complicated nature of the technology affects the knowledge transfer and makes it much more difficult to find quid pro quo "nuggets," particularly in jurisdictions of other countries. Mechanisms facilitating identification of those special technologies must be improved. Various options available to address this problem on the U.S.-Canada bilateral level are presented. Similarly, possibilities for cooperative research and development in robotics-related areas are addressed.

Introduction

Background

In many countries today, robotics is seen as a key emerging technology with application in both the military and civilian environment. There are benefits, however, to leveraging sources for that technology outside national boundaries. The benefits of leveraging foreign technology include

- it is in keeping with today's increasing environment of multinational military ventures;
- it allows you to capitalize on the advanced technology developed by allies;
- it can make programs more affordable by spreading costs among a number of partners; and
- it avoids duplicative research and development (R&D) expenditures.

New economical realities are also resulting in some new constraints and some new opportunities in cooperative developments. The quid pro quo stand is necessary to effect exchange of technology and to prevent its unidirectional flow. Concurrently, fewer available resources call for the most cost-effective and pragmatic approach to development. This implies the need to use the best and most suitable technologies, regardless if their origin is national or foreign, thereby calling for effective cooperative international efforts.

Currently, no formal bilateral arrangements exist between Canada and the United States specifically in the robotics area. However, the 30-year-old Defence Development Sharing Agreement (DDSA) between the U.S. and Canada is very broad and could therefore facilitate a type of cooperation about specific topics in robotics R&D. Additionally, a multinational information exchange about operator-robot interaction (ORI) has been in place for the last 4 years through the NATO, AC/243 Defence Research Group, Panel 8/Research Study Group (RSG-18). Its membership included Canadian and the U.S. representatives, along with a number of European countries, and it enhanced familiarization with robotics in the member countries. Similarly, the Technical Cooperation Program (TTCP) is a multinational activity involving Australia, Canada, United Kingdom, and the U.S. Some information exchange took place also under this program through the activities of the Subgroup W, Technical Panel WTP-6.

Objectives

This paper focuses on bilateral cooperation opportunities in defence robotics. Various applications of robotics in support of defence requirements in each country may be identified and pursued. The unmanned ground vehicle (UGV) theme may serve as an example for land operations requirements. The report describes a process for identifying foreign technology opportunities and discusses various facilitating mechanisms available on the U.S.-Canada bilateral level. Specific ongoing programs and possibilities for collaborative research and development in UGV-related areas are briefly outlined.

The Quest for Technology "Nuggets"

Obstacles

The search for technology nuggets (i.e., foreign technology concepts/components/methodologies that could potentially enhance a nation's robotics program) is impacted by several factors:

- The first question to be resolved by each nation is to determine where to start. Many countries have a robotics program in progress, but which ones have something to offer that the others do not? Limitations of resources demand that this question be answered in some other way than by visiting each country. Then again, robotics activity may be spread within a country, that is, among the universities, Government laboratories, and various commercial ventures.

- Experience has taught us that technology nuggets exist at the subsystem/component level. For example, many countries have UGV programs, but these may differ in their technical approach. One country may have a UGV obstacle-detection and-avoidance package that, at a quick glance, appears to have the same performance characteristics as those in most other UGV programs. However, this particular obstacle-detection and-avoidance package may have a particular unique design approach that would classify it as a nugget.

- Robotic solutions are used in a very broad spectrum of applications ranging from manufacturing in highly structured environments to hazardous material handling, underwater operations, space activities, or UGV operations involving what is commonly called "unstructured" environments. Robotic solutions are also used in a broad range of other applications that are not explicitly identified as a part of the robotics domain (for example, simulation technology). Consequently, some robotic technological nuggets may be difficult to locate and identify because of their "remoteness." Nevertheless, they may be most suitable and most desirable for "transplanting" to applications in other than the original environment or application. Therefore, a broad access to the full spectrum of applications is most desirable.

- Robotics is a multidisciplinary domain with numerous technical subspecialties. The complicated nature of robotics affects knowledge transfer and makes it much more difficult to find technology nuggets. It is impossible to take a full battery of experts to another country or even to an in-country convention, to cover all topics of interest (e.g., the computer scientist to cover control architecture, the sensor specialist to obstacle/target detection technology, or the mechanical engineer who specializes in manipulator design).

- How do you validate a technology? Another country can claim to have a UGV with a road-following system that allows travel down a paved road at speeds exceeding 60 mph. Is this claim valid? Are there certain conditions which seriously impact its performance? For instance, the same UGV may be able to travel at 60 mph down a paved road but in rain, might be stopped altogether by rain puddle reflections that cause image-processing confusion. These questions underline the need for dynamic equipment demonstrations, standard performance metrics, access to valid performance data, and candid discussion with the researchers of origin by experts in the topic.

Facilitating Mechanisms

Conferences

At both the national and international level, robotics-oriented conferences exist that can facilitate the nugget search process. Examples include the International Robotics and Vision Conference next scheduled for 1995 in Detroit, Michigan, the 24th International Symposium on Industrial Robots held in Tokyo, Japan, October 1993, the 20th Annual Technical Symposium and Exhibition sponsored by the Association for Unmanned Vehicles Systems (AUVS) that occurred June 1993 in Washington, DC, IEEE, The International Society for Optical Engineering (SPIE), and a number of other bodies sponsor regularly a number of excellent technical conferences of importance to the robotics field. NATO-sponsored conferences such as the DRG Seminar on Robotics in the Battlefield, held in Paris in 1992, provide an overview of the international defence robotics scene. The workshops about military robotic applications held biannually in Canada facilitate an overview of the Canadian advances in the field.

Reports/Magazines/Periodicals

Published conference proceedings can sometimes substitute for actual attendance. Such publications can also fill information gaps that were missed, even if the conference was attended. For example, the 20th Annual AUVS Technical Symposium and Exhibition (mentioned before) published a 1148-page proceedings manual and a full list of attendees, with addresses and telephone numbers, that would be very useful in follow-up actions.

Other examples of reports/magazines/periodicals are *Robot Times, Industrial Robot International Quarterly, Journal of Robotic Systems, Military Robotics, AMSE Journal of Dynamic System Measurement and Control, IEEE Transactions on Robotics and Automation, International Journal of Robotics Research, and Robotics World.* Many times, however, relevant articles exist in publications that are not directly related to robotics, making the literature search process very difficult. These more esoteric sources could have a military orientation, such as *National Defense*, the journal of the American Defense Preparedness Association, or a peripheral technology orientation, such as the *Proceedings: Image Understanding Workshop* sponsored by the Advanced Research Projects Agency.

University Research Centers

Universities are typically working on the cutting edge of technology, but the research they conduct is normally basic in scope and not ready for integration into functional prototypes for some time. Reports about university results are generally easy to access (i.e., not restricted as proprietary data); however, many academic institutions in both the U.S. and Canada are involved in robotics research, and it is difficult to keep track of all their activities.

Personnel Exchange Programs/Visits

The exchange of scientists and engineers is possible through some of the international agreements/forums described below. Such exchange programs help foster collaboration and, more important, are one of the best ways to identify specific quid pro quod opportunities.

Trade Shows/Demonstrations

Trade shows offer the opportunity of conducting quick, usually static, surface-level searches for technology nuggets. Examples include the 20th - Annual AUVS Technical Symposium and Exhibition mentioned before.

Demonstrations are preferred to trade shows because demonstrations typically offer dynamic performance data in a quasi-realistic operational environment. Demonstrations are key to addressing the technology validation obstacle discussed before. A joint U.S./Canadian demonstration of countermine-robotics is being considered by both nations. Such a demonstration tends to supplement each other's program with the knowledge and capabilities of the other country and promotes concerted action to identify and close important gaps in the collective technology base.

International Agreements/Forums

The most active U.S./Canada agreement in use by military robotics interests is the Defence Development Sharing Agreement (DDSA). DDSA working groups are responsible for nominating candidate R&D projects to be considered for joint U.S.-Canadian funding. Currently, discussions are progressing that have a working objective of developing a joint U.S.-Canadian engineering development program about a tactical unmanned ground vehicle.

Another possible avenue for collaborative activities is the Technical Cooperation Program (TTCP). TTCP affects the interchange of information, personnel, and material for the participating countries (i.e., Australia, Canada, United Kingdom, and U.S.) with each other's defence technology programs in

- Basic research
- Exploratory development
- Advanced technology demonstrations

A proposed project is currently being staffed through TTCP Working Technical Panel-6 entitled "Operational Assessment of Battlefield Robotics." The objective of that project is to identify, evaluate, and recommend methodology that will be used to define, with confidence, the utility of robotics on the battlefield. The U.S. and United Kingdom have agreed to conduct the operational assessment in that project, with Canada and Australia providing technical input and review.

A robotics-oriented Information Exchange Annex between the U.S. and Canada is currently in staffing. This Annex will provide for the exchange of data about research and development program descriptions and experience in the field of robotics for the planning, design, and operation of robotics systems for military forces. DEA coverage includes teleoperated and autonomous robotic vehicle platforms and manipulators, for both weapons system and materials handling applications, as well as the component technologies of robotic systems such as high payload-to-weight ratio manipulators, artificial intelligence aspects of robotic systems, and simulation techniques used to assess the military effectiveness of robotic systems and component technologies.

A Proposed Process

The obstacles to the search for foreign technology nuggets and existing mechanisms for producing international cooperative opportunities suggest the following five-step process.

Phase 1 (initial search)

The initial (macro) search process aims at narrowing down the robotics technology nugget search process down to specific countries and perhaps specific general subject areas (e.g., manipulators, navigation, autonomous driving). Mechanisms that best support this objective are

- Review of reports, magazines, and periodicals
- Attending conferences, symposia, and so forth
- Attending trade shows

Phase 2 (focused search)

The focused (micro) search process centers on further narrowing the search to a particular activity of interest within a specific country. At this point, the problems associated with the multidisciplinary nature of robotics typically begin to surface. If the target nugget is technically diverse, then it may be necessary to try to bring enough technical detail back for further analysis by subject experts. Mechanisms that best support the focused search phase are

- Personnel exchange programs or visits
- Attending trade shows and demonstrations

Sponsorship of an international agreement or forum may be important at this phase, particularly if there is a requirement to bring back technical detail for further analysis by subject experts.

Phase 3 (classification)

Having located a potential technology nugget, it becomes necessary to validate its worth to one's robotics program before considering acquisition possibilities. As suggested before, this might require convening a panel of experts who would cover the complete range of technical facets of the nugget in question. At this time, preliminary discussions of quid pro quo would also be appropriate.

Phase 4 (acquisition)

Having decided in the previous phase that the nugget is worth pursuing, the issue then becomes how to acquire it. It may be as simple as outright purchase, or especially if advanced technology is involved, a quid pro quo would be appropriate. Again, sponsorship through an international agreement or forum may be important in this phase.

Phase 5 (evaluation and integration)

Following acquisition, the last logical phase would be hands-on evaluation of the nugget and subsequent integration into a national robotics program.

Conclusions

This paper looks at the quest for foreign technology nuggets in robotics, discusses the obstacles, facilitating mechanisms, and a suggested process. This is not to say, however, that all has been resolved. On the contrary, the nuggets continue to elude the search process; there are so many countries, industries, and universities throughout the world doing work in robotics that it is difficult to accomplish any sort of comprehensive search without using significant resources. The problem is compounded by the fact that nuggets often occur on the periphery, that is, exist in another technology area but could be of significant benefit to a robotics system. The whole process might be simplified if something like the Paris Air Show existed for robotics, but there is nothing comparable. Perhaps as robotics matures and applications abound, increased interest will bring with it the resources that will make the quest easier.

References

AUVS-93 Proceedings Manual, [1993], Washington, D.C., Association for Unmanned Vehicle Systems.

Beni, G. and Hackwood, S. (Eds.). [1993, February], *Journal of Robotic Systems*, NY: Wiley Publishers.

Bradley, B. [1993, July] *Proposed TOR for TTCP KTA 6-21: Operational assessment of battlefield robotics*, [Viewgraphs] Aberdeen Proving Ground, MD: U.S. Army Materiel Systems Analysis Activity.

Burnstein, J. (Ed). [1993] *Robot Times*, Ann Arbor, MI: Robotic Industries Association.

France, J. (Ed.). [1993, Summer] *Robotics World*, Atlanta, GA: Communications Channels Inc.

Loughlin, C. (Ed.). [1992] *Industrial Robot International Quarterly*, Singapore: MCB Publications Ltd.

Phelps, R., Shoemaker, C., and Hodge, D. [1993, January] *International Robotics Strategy*, Aberdeen Proving Ground, MD: U.S. Army Research Laboratory.

Proceedings: Image Understanding Workshop, [1993, April] Washington, DC: Advanced Research Projects Agency.

Proceedings from the 31st DRG Seminar on Robotics in the Battlefield, NATI Technical Proceedings, AC/243-TP/3, Paris, [1992].

Proc. DREP/RRMC Military Robotic Applications Workshop, Victoria, B.C., [Aug 1987], DREP Special Report 87-1.

Proc. Second Workshop on Military Robotic Applications, Kingston, Ont., [Aug 1989], DCIEM Special Report 90-1.

Proc. Third Conf. on Military Robotic Applications, Medicine Hat, Alb., [Sep. 1991], DRES Special Publication 149.

Progress in Conventional Weapons Technology Through International Cooperation, Collaboration and Exchange, TTCP Subgroup W Brochure.

906216

U.S. Unmanned Ground Systems - A Strategy for System Development and Fielding

Mark L. Swinson, Ph.D., P.E.
Lieutenant Colonel, U.S. Army 04a
Unmanned Ground Vehicles/Systems Joint Project Office
AMCPM-UG, Redstone Arsenal, Alabama, USA 35898-8060

#149697

Abstract

There are few that would argue that unmanned ground systems have truly begun to realize their full military potential, or perhaps even to have that potential fully realized. The reasons are many and diverse, but not insurmountable.

Recent changes within the Department of Defense (DoD) Joint Program for Unmanned Ground Vehicles (Robotics) are intended to overcome a number of these impediments to system development and fielding. Perhaps most striking is the broadening of the scope of responsibilities for the Joint Project Office (JPO), located at Redstone Arsenal, Alabama.

This paper will detail some of the changes, as well as their anticipated consequences. Further, an overall "robotics strategy" will be articulated. This strategy should help to clarify the framework for the program over the next few years, which encompasses not only the fielding of several unmanned ground systems to the U.S. Armed Forces, but also supports the development of a robust, technology infrastructure that will support U.S. efforts in the field of unmanned ground systems and robotics for many years to come.

Background

In fiscal year 1990, in response to a request from the U.S. congress, all DoD advanced development projects related to ground vehicle robotics were consolidated in a single program element (PE 0603709D) under the Office of the Secretary of Defense (OSD). This program element is called the OSD Joint Robotics Program (JRP). As part of this consolidation, existing and emerging requirements were reviewed. Several were terminated, while the rest were generally restructured to produce a more focused and cost-effective program. Since then, individual projects have been tied together into a much more coherent acquisition process. In addition, a robotics technology maturation effort was initiated to establish a national research and development base for unmanned ground vehicles (UGVs). Hence the current program consists of advanced technology development at the component level and the advanced development of system prototypes. This program is reported to Congress in an annual Unmanned Ground Vehicle Master Plan (UGVMP).

Program Scope

The robotics program has been structured to progress from teleoperation--where a remotely located human directly controls the UGV--to reliance on autonomous onboard performance of various UGV functions with the operator in a supervisory role, and possibly able to supervise multiple UGVs concurrently.

The current UGV investment focuses on:

- Near- and mid-term advanced system development projects that have strong Service support with requirements that are approved, or well along in the approval process.
- Overcoming technological barriers that impede the development of more autonomous UGVs.

The robotics program builds on the research and exploratory development carried out by the Services and defense agencies, integrating demonstrated component technologies into prototype systems that promise high operational payoffs. User experimentation with existing UGV systems provides a firm basis for concept formulation. Prototype development and technical testing coupled with expanded user experimentation provide the necessary basis for system development and acquisition decisions.

Although unmanned aerial vehicles (UAVs) are not a part of the UGVMP, mutually enhancing operational interactions between UGVs and UAVs, interoperability of remote command and control stations, and commonality of architectures are being pursued. Most recently the Unmanned Ground Vehicle/Systems Joint Project Office (UGV/S JPO) conducted a UAV/UGV interoperability demonstration at Redstone Arsenal, AL. Various technologies--including artificial intelligence (AI), fuzzy logic, and advanced image processing--that other programs are developing are being used for UGV systems. Conversely, the projects within the JRP are being applied to other program areas such as the physical security program, environmental clean-up, and obstacle breaching.

Basis for UGV Systems Planning

There are strong reasons for believing that the structure and operations of future forces will depend heavily on robotic systems:

- A variety of potential applications of robotics to land operations can increase efficiency and safety. These include reconnaissance and surveillance, target acquisition and engagement, logistics support functions, minefield detection and neutralization, obstacle breaching, explosive ordnance disposal, physical security, and operations in contaminated environments. (The threat of encountering chemical and biological weapons in Third World conflicts is growing rapidly.)
- U.S. Forces size will be substantially smaller, but without commensurate reductions in responsibility.
- Environmental restoration of existing and retired defense sites will be high priority requirement for many years to come.
- Technology and subsystems necessary for many of these applications are developing rapidly. Modern sensors, computers, software, and communication links can acquire, process, and transmit data far beyond the capabilities of individuals restricted to manual operations.
- Robots and robot-like devices continue to grow in commercial importance with applications that range from cruise control of automobiles to complex autopilots and from manufacturing to medicine.

Because of these and other factors, reinforced by the experience in Desert Storm, there is rapidly increasing user recognition of requirements for UGVs. A number of lessons learned from the recent conflict in Southwest Asia reinforce the view that automation and robotic systems will play a major role in realizing the force multiplication required to offset continuing reductions in land forces:

- Robotic systems, in this case UAVs, were widely used in combat for the first time.
- Land forces confronted the immediate threat of chemical weapons.
- A hurried request for remotely operated mine-clearing tanks and shallow water landing craft was made.
- High technology weapons demonstrated the effectiveness of autonomous guidance.
- Desert Storm set a standard of minimal friendly casualties against which the results of future conflicts will be measured.

The potential applications and payoffs of UGV systems are apparent and today's sensors, computers, and communication systems, coupled interactively with humans, can meet the requirements for many of these applications. However, for many other applications the software that is required for reduced bandwidth communication and for approximating human capability to integrate data and exercise control authority does not exist. Developing such software for future autonomous and semi-autonomous systems is one of the major challenges. Teleoperation avoids the severe software challenges of autonomous operation but involves significant communications, man-machine interface, and human factors requirements. Teleassistance, i.e., the automation of subfunctions such as steering, brake, and throttle control, can reduce these requirements significantly. Teleoperation is feasible today; teleassistance is rapidly proving its feasibility; except for highly structured applications, fully autonomous operation is not yet feasible, but technical progress is being made.

Given the status of robotics technology and user requirements, current U.S. DoD planning for UGV systems stresses two types of system development. The first type involves UGVs that must operate on the battlefield with other land combat systems and forces. The Tactical Unmanned Ground Vehicle (TUGV) development is the primary thrust of the UGV program in this category. The second type involves relatively narrow, but important and cost-effective, applications that have more limited and achievable requirements for automating human control functions. These narrower projects develop specific robotics techniques and also support the goal of gaining user acceptance of robotic systems and public confidence in their safety and effectiveness.

The UGV program strategy is based on a coordinated evolution of demonstrated capabilities and user requirements. During the last 3 years, teleoperations and teleassistance for battlefield environments and more autonomous technologies for structured environments were emphasized, together with extensive user opportunities to gain experience with prototypes. This is proving effective in stimulating specific requirements and new applications. In the mid-term, supervised robotics demonstrations will be expanded for navigation and reconnaissance, surveillance, and target acquisition (RSTA) functions. In the far term, highly robotic systems based on artificial intelligence will be developed not only for autonomous TUGVs but also as upgrade options for manned systems. Far-term robotics research is being conducted mainly by the Advanced Research Projects Agency (ARPA).

The UGV plan has been structured to maximize future use across the Services, to focus limited resources on existing and emerging requirements, to ensure close coordination among several UGV projects, and to preclude duplicative activities. Technology programs have been focused on system applications. Specific programmatic plans, described below, have been developed to ensure the timely demonstration of appropriate technology and the integration of demonstrated technology into system development.

Considerable attention has been given to fostering organizational relationships (among DoD components and between them and other agencies, universities, and industry) that will assure technology transfer, promote dual-use, and maximize the productivity of the funds expended under the JRP. International cooperative opportunities are also being pursued. These relationships include:

- Development of the TUGV under a Memorandum of Agreement (MOA) between the U.S. Army and U.S. Marine Corps.
- Establishment of the UGV/S JPO as the focal point for coordination of UGV projects.
- Collaboration among the advanced system development and technology projects for demonstration and integration of technology prior to engineering and manufacturing development (EMD).
- Development of robotic UGV navigation technology under an MOA between ARPA and the Office of the Undersecretary of Defense (Acquisition and Technology)/Tactical Systems (OUSD(A&T)/TS) that incorporates technology transfer from academia to industry and Government.
- Cooperation between the UAV and UGV project offices in the development of employment concepts and common control architectures.
- Pursuit of joint development opportunities between DoD and other departments, specifically the Department of Energy (DoE) and the Department of Transportation (DoT).
- The UGV/S JPO and the Canadian government are actively investigating cooperative development and defense development sharing opportunities.

Program Summary

The overarching concept for the UGV program is a "family of UGV platforms" which is developed with maximum integration among the individual UGV system projects. The UGV/S JPO has been assigned the responsibility and authority for executing the required integration. The goal is to field a family of UGVs that cover a spectrum of important applications. Most of the platforms will be existing vehicles. The platform size of each UGV will be appropriate to the application. The various subsystems (sensors, communications, control, etc.) will be based on shared technology and maximum commonality. Within a size category, even the platforms can often be common across applications.

The following paragraphs briefly discuss the individual items in the UGV family:

- The TUGV is a joint Army and Marine Corps effort to develop UGVs for battlefield operations. The TUGV provides the human operator with a standoff capability to acquire real-time acoustic and imagery intelligence from a location that is relatively safe from both enemy fire and fratricide. Its overall objective is to increase the effectiveness and survivability of combined armed forces by extending the control radius of the soldier on the battlefield. The TUGV task is characterized by an unstructured, dynamic, and hostile environment which includes active human opposition. This project was initiated by Congressional request in FY 1990. The near-term objective is to field a teleoperated/teleassisted TUGV with the remotely located operator capable of performing RSTA functions, chemical detection, and laser designation. The near-term program emphasizes infantry use and includes extensive user familiarization with prototypical UGVs to assess operational benefits and liabilities in preparation for Milestone I (MS I) and to assist in refining requirements before entering EMD in FY 1997. The current plan is to procure about 1,100 TUGVs.

- The Remote Ordnance Neutralization System (RONS), now in advanced system development (ASD), is an explosive ordnance disposal (EOD) UGV acquisition project with both military and civil application. The task is characterized by an unstructured, relatively static indoor or outdoor environment. The RONS operational mode strongly resembles that found in the conventional manned bomb disposal task, with the RONS system providing standoff for the bomb disposal technician. This UGV system will provide EOD personnel safe separation from hazardous accident/incident sites where explosive, chemical, or radiation risks are present. RONS is a remotely operated mobile platform with closed-circuit TV, a six-degrees-of-freedom manipulator, and a suite of EOD-unique tools with automated tool exchange. Transition to EMD is planned for FY 1995 with production commencing late in FY 1998. The current plan calls for procurement of about 200 RONSs.

- The Mobile Detection Assessment Response System-Exterior (MDARS-E) has just begun. It is part of the physical security equipment (PSE) program funded under PE 0603228D. MDARS-E will be a highly autonomous UGV that performs surveillance patrols, intrusion detection, barrier assessment, and theft detection. It is intended to protect storage facilities and their contents. Because coordination of the MDARS-E and TUGV projects has important potential benefits to both projects, the process of coordinating and integrating the two projects has begun. The first step, a Memorandum of Understanding (MOU) between the respective project offices, has been completed.

- The Infantry School TUGV requirement calls for a small vehicle. The Armor School is also considering a requirement for a UGV, but is not yet fully defined. The small vehicle concept for TUGV may be compatible with this emerging requirement depending on the UGV missions that are specified by the Armor School.

- The Field Artillery School is beginning to consider a UGV requirement for the forward observer mission to provide sustained forward observation and control of artillery fires at extended ranges. Again, the UGV size (small or medium) will depend on mission requirements and size/weight/power requirements of the mission module.

- The Engineer Vehicle Teleoperation Capability (EVTC) project is a requirement that has emerged since last year's UGVMP. The concept is a generic, modular kit that could be installed on a variety of existing manned vehicles, mineclearing tanks, earthmovers, bridging vehicles, amphibious vehicles, and bulldozers. Without interfering with manned operation, the kit would allow remote, unmanned operations as circumstances require.

- The Robotic Excavation Vehicle (REV) project is responding to a requirement for range clearance by adapting the equipment and technology developed under the Rapid Runway Repair (RRR) program.

With respect to development of new supporting technology, the emphasis is on DEMO II now that DEMO I has been completed.

- DEMO I--This part of the UGV Technology Enhancement and Exploitation (UGVTEE) effort reached a milestone in April 1993 with an ambitious series of demonstrations at Aberdeen Proving Ground (APG). DEMO I emphasized the conversion of existing manned vehicles to robotic accelerate the maturation of the most critical teleassistance technologies. The DEMO I work was completed in FY 1993 with the technology being transferred to system development projects.

- DEMO II--This is the other major effort within the UGVTEE. It started in 1991 with the objective of developing, within 5 years, those navigation and other technologies that are necessary to move UGV capabilities from teleoperation to supervised robotics. DEMO II emphasizes autonomous navigation under battlefield conditions, RSTA functions while the UGV is moving, automated communication with other vehicles and with the supervising command and control network, and workload partitioning among vehicles.

UGV/UAV Technical Integration Demo

As mentioned above, one area of significant interest for the JRP is the commonality and interoperability of UAV and UGV systems. While the operational integration of these systems has been demonstrated before in simulated tactical scenarios, such as the combined UAV/UGV demonstration hosted by the UGV JPO and held at the Marine Corps Base Quantico, Virginia, in October, 1990, no technical integration demonstration had heretofore been accomplished.

On 13 September, 1993, such a technical integration was demonstrated at Redstone Arsenal, Alabama. The UGV/S JPO and the UAV JPO jointly hosted a combined demonstration of some of the advantages and opportunities that can accrue to the integration of aerial and ground based unmanned systems. In this demonstration a Surrogate Teleoperated Vehicle (STV) UGV was remotely driven to an observation point using a covert, fiber optic data link. A hand-launched UAV (Pointer) was then deployed. (See Figure 1.)

The UGV was able to observe and track targets within its field of view. However, the UAV was able to provide additional, lower resolution video information about enemy activities beyond the tree line or battlefield obscuration. Furthermore, since control of the Pointer was handed off to the STV (that is the Pointer was actually flown via the STV controller), an enemy with direction finding equipment might be able to locate the Pointer's control signal source (the STV), rather than the scout team. Thus, only the unmanned platforms were put at risk. Furthermore, the effective range of the Pointer is automatically extended by the distance from the operator to the UGV. (see Figure 2.)

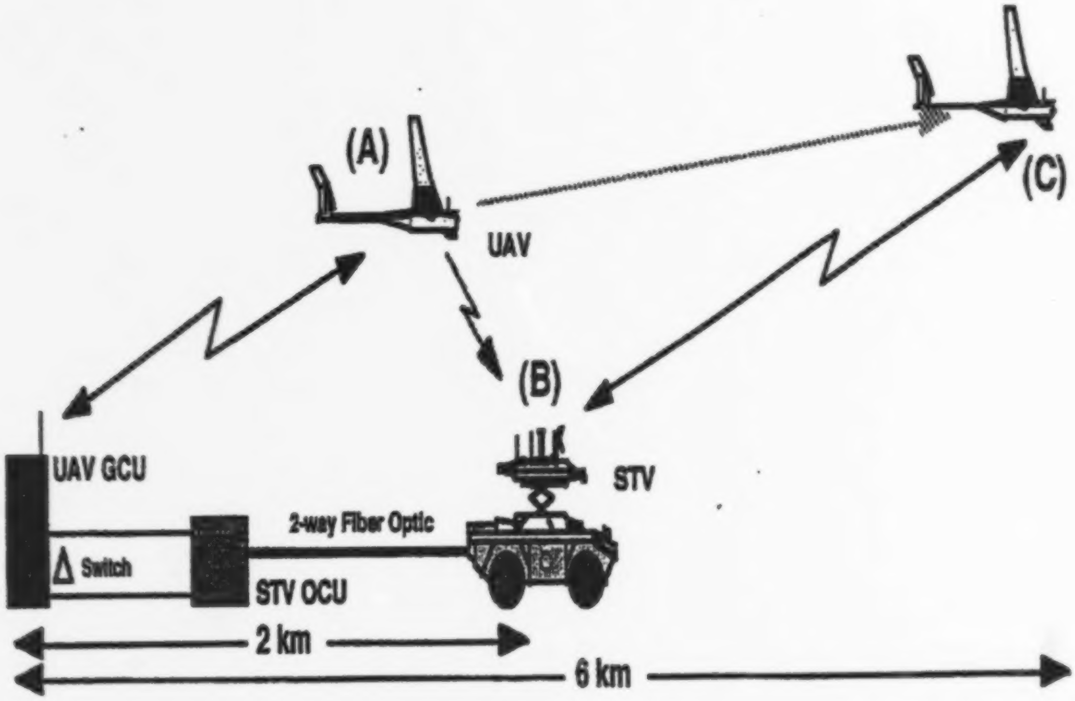


FIGURE 1

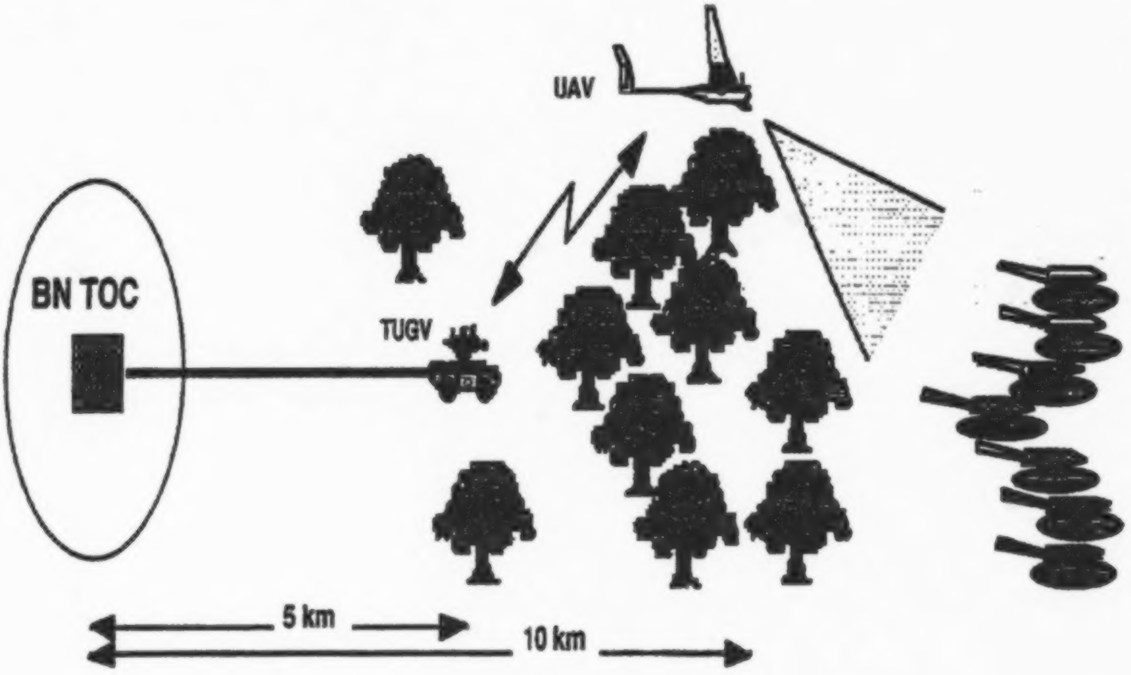


FIGURE 2

Additionally, the video images relayed from the Pointer were superimposed onto the STV video display (picture in a picture). Hence this system provided the human operator with the ability to simultaneously view an object from both a top and side perspective. This greatly enhanced the operator's target identification capability, and suggested significant potential for reducing the changes of fratricide due to target misidentification.

Conclusion

The Joint Robotics Program appears to be well on its way to bringing to reality the promise of unmanned systems for our armed forces. The program is highly integrated, including technology development, requirements development, and system development. With continued hard work, these efforts will bear fruit in our quest to take our forces, where ever possible, out of harm's way.

90C98C

DND Knowledge-Based Systems and Robotics Workshop, November 1993

Defence Robotics in Canada; an Overview

Julius J. Grodski
Defence and Civil Institute of Environmental Medicine
P.O.Box 2000, Downsview, Ont., Canada, M3M 3B9
email: jul@dciem.dnd.ca

#149C98

Abstract

Robotics activities within the Department of National Defence (DND) constitute the primary focus of the review. Existing specific operational applications of robotic systems in land, air, and sea environments are identified. Some of the future requirements of the Canadian Forces in robotics are addressed by the Defence Research Establishments (DRE's) in cooperation with the Canadian industry and universities. Themes of robotics-related research at the DRE's are outlined and some details are given on the R&D pursuits undertaken and coordinated in this area. DND also maintains a close contact with the university and industry - based research in robotics through its membership in PRECARN, a non-for-profit organization which fosters precompetitive research into artificial intelligence and robotics. The major robotics projects carried out through PRECARN coordination are briefly described. The Military Colleges also contribute to the DND robotics technological base. Robotics activities at the Military Colleges are summarized. The Artificial Intelligence and Robotics Steering Group was organized as a consequence of the Second Workshop on the Military Robotics Applications. The Group's role and objectives are outlined. The paper provides extensive references to the reviewed activities.

General

The position of the National Defence Canada with respect to robotics was restated recently and it indicates that:

> "No explicit national goals in robotics have yet been formulated nor is it expected that any will be deemed necessary. The technology is and will likely be used when it proves itself to be effective in specific applications. Nevertheless, robotics related Research and Development (R&D) will continue to be an important area for Canada and for the Department of National Defence (DND).
>
> As a member of PRECARN, Chief of Research and Development (CRAD) has opportunities for leveraging Canadian R&D in Robotics and Artificial Intelligence (AI). This, together with our active participation in international information exchanges, provides an effective network for R&D in robotics and related areas.
>
> The recent Technological Trends, Threats and Opportunities Workshop identified autonomous systems as important to future Canadian Forces (CF) requirements. The relevant R&D issues included cost effectiveness; surveillance and minefields -- possibly using robotics and Remotely Piloted Vehicles (RPVs); power sources, materials and miniaturization; guidance and control and navigation exploiting AI/expert systems; bio-sensors and bio-regulators, and the human-machine interface.
>
> In terms of our future strategy, the focus remains one of maintaining an advanced level of knowledge for advising the CF by leveraging international and national capabilities so that we can exploit these technologies in order to respond to future CF requirements. In addition to this direct potential for military applications, the laboratory work on robotics continues to be an element in furthering information exchange and industrial participation. "

The general theme of robotics for defence has been explored in keynote presentations[1-3] at the preceeding three biannual Military Robotics Applications Workshops held in Canada. Strategic implications of robotics[4] and social and economical factors of robotics systems[5] were also explicitly addressed at the Workshops.

Robotics in CF Operational Applications

Land

The land element of the CF uses the Remote Mobile Investigator (RMI-3) teleoperated system in the Explosive Ordnance Disposal (EOD) operations. RMI-3 is equipped with a simple two-link arm with a claw, mounted on a mobile platform operated through a cable connection from a control station. A hinged, 6-wheel base with two independent drives gives it highly flexible mobility characteristics. The system includes various attachments extending its range of applicability.

It is expected that CF will add another teleoperated system to its inventory in the form of the "Badger" target vehicle, described further below.

An in-depth survey of technologies that can be applied to all levels of robots from teleoperated to autonomous in the land environment was carried out by a contractor[6]. Volume 1 of the Phase I report dealt with the army operational environment and the testing environment as well as the army doctrine. Volume 2 contains sections on technology and information on robotic systems under development. In Phase II report, 31 possible applications were identified and analyzed in terms of operational effectiveness, cost, risk, etc. Three applications were selected from the long list and they were examined in detail in Phase III report. The report also provides order-of-magnitude cost estimates for development of demonstration systems.

Sea

Underwater and surface vehicles pertain to different operations in the sea environment. Underwater Remotely Operated Vehicles (ROVs) are increasingly used to address a number of objectives which previously were normally resolved almost exclusively by a diver. ROVs are frequently used for underwater observation, cable laying and repair, weapon recovery, salvage support, and increasingly for various mine countermeasure (MCM) objectives.

CF currently holds two types of underwater ROV systems in its inventory in addition to two manned submersibles, capable of deployment of diving teams and deep diving for observation and operations involving the use of on-board manipulator capabilities. The unmanned ROVs are the Deep Access Reconnaissance Television (DART) and MANTA:

- DART has been designed as an underwater observation tool for the construction and offshore industry for operation at depths of up to 360 m. The CF version consists of a self-propelled closed circuit television system with lighting, a high-frequency sonar, and a pivoting hook for payload delivery/drop-off. The vehicle is very compact and light weight, providing for easy transportation in a minimum of space.
- Manta has been developed in Canada for subsea inspection and surveillance. It is equipped with a sonar, and video and still cameras. The active sensing data is transmitted through a fiber optic cable to a mother vessel. In its current application, it is towed by a 75 ft wooden vessel and it is limited by the cable length to 125 m maximum depth. Manta is the only full time MCM asset doing detailed route survey.

Other systems not in the inventory of the CF but available from Canadian manufacturers for underwater defence objectives include the "Dolphin" semisubmersible and the "Trail Blazer" ROV.

Air

CF has no unmanned air vehicles (UAVs) in its inventory. However, significant developments have been carried out in this area through DND-sponsored activities to develop the CANADAIR CL-227 UAV system for Army and Navy applications. Its major advantage is its vertical take-off/landing capability, low profile and signature.

An earlier land-based version of the UAV was jointly tested by the Canadian and US Armies in 1989-1990. More recently, a version of maritime CL-227 UAV System has undergone evaluation on board USS Doyle, as part of the US Maritime Vertical Take OFF and Landing UAV System (MAVUS) program. Canada, the Netherlands and the UK participated in the trial.

Chief of Research and Development (CRAD)

CRAD is the Branch of National Defence Canada which administers defence research and development activities, including robotics, through in-house activities at a number of Defence Research Establishments (DRE's) across the country and through contracted activities carried out in the Canadian industry, universities, and various other organizations. The theme of robotics has been actively pursued at two DRE's: the Defence and Civil Institute of Environmental Medicine (DCIEM) in Toronto, and the Defence Research Establishment Suffield (DRES) in Suffield, Alberta. Some activities have also taken place at two other DRE's: Defence Research Establishment Pacific (DREP) in Victoria, B.C., and Defence Research Establishment Valcartier (DREV) in Quebec.

DCIEM

DCIEM's mission is to enhance the effectiveness and ensure the health and safety of the human in any human-machine system or adverse environment. Robotics at DCIEM focuses on the unstructured environments as the dominant factor shaping application of the multidisciplinary technology for military[A1] and dual-use operations. This outlook has narrowed the scope of the activities to telerobotics where the human presence in the control loop is essential, at least at this stage in the development of the technology. Specifically, robotics at DCIEM concentrates on the development of subsystems for teleoperated and telerobotic systems and evaluation of specific Operator-Robot Interaction (ORI) issues. The five themes of the activities are briefly described:

Telepresence: sensing and display

The main thrust of the telepresence activity is in vision, and specifically in the stereoscopic vision area. The main achievement was the development of the Augmented Reality through Graphic Overlays in Stereovideo (ARGOS) capability[A2-A8] and its patenting[A9]. The work resulted in the first-ever stereoscopic video contribution/publication[A10]. ARGOS has been demonstrated in conjunction with various platforms, from simple Amiga PC to IRIS workstation. An overview of real and virtual world stereoscopic displays for teleoperation was carried out[A11]. Learning effects in telemanipulation were studied[A12,A13] and a range of human factors topics were studied[A14-A21]. Benefits of stereovideo were also investigated in experiments[A22] using naive and expert operators of teleoperated Explosive Ordnance Disposal (EOD) vehicles. Applications of Augmented Reality for human-robot communication and related topics were examined[A23-A28]. Initial work was also carried out on force and tactile sensing and feedback.

Actuation

A pneumatically-operated Robotic Muscle-like Actuator (ROMAC)[A29-A31] was developed through DCIEM efforts into a strong, effective, high force-to-weight ratio device. This is a unique actuator which in an antagonistic pair arrangement provides the means for independent control of position and compliance of a robotic link. ROMACs have been fully characterized. A family of joint control algorithms with special joint behaviour characteristics facilitate application of ROMACs in robotic devices such as manipulators and grippers. High strength, flexibility and endurance tendons compatible with the actuator and ROMAC optimizing cams have been also developed. A unique ROMAC multifunction gripper is currently under development. The developed force and tactile sensing capabilities are directly applicable to the ROMAC-based gripper master/slave system.

Biologically-based control capabilities

Various biologically-based control capabilities have been developed through DCIEM activities:

- myoelectric control of ROMAC actuators[A31-A34] whereby robotic manipulators may be controlled directly from human muscles;
- gaze-contingent display[A35] which is light, simple, inexpensive, and allows head movement, and
- parametric control system[A36-A38] comprising an adaptive controller in form of a neural network which eliminates the need for inverse-kinematics calculations which are difficult to implement in real-time, e.g. to control flexible robotic arms.

Force and motion generation

Force and motion generation has been studied with the aim to develop a self-contained, simple but effective simulator for shoulder-supported rocket launchers. Related 6D motion analysis was carried out to achieve correlation of different measuring devices.

Mobile teleoperated systems

Developments on mobile teleoperated systems focused predominantly on enhancements of a commercially available EOD teleoperated device: Remote Mobile Investigator (RMI-3). The various enhancements[A39-A41] encompass:

- applied human factors criteria to redesign the RMI-3 controller;
- multiplexed control signals;
- demonstrated fiber-optic communications link between the remote and control stations;
- implemented control/audio/video multiplexed co-ax communications link;
- designed and demonstrated extended range for the robot through the use of a constant tension cable pay-out/pick-up system with the spooler on the remote unit;
- designed and demonstrated collapsible/extendible cable mast preventing cable entanglement.

DRES

DRES is involved in the development of robotic vehicle technologies[B1-B27]:

ROBOT-X

Designed to exercise point defense missile system at an affordable cost, ROBOT-X provides a maneuverable, rocket-boosted aerial target that requires no in-flight operator control and limited ground support. The target system provides such features as high subsonic speeds, low-altitude hold, pre-programmed maneuvering and over-the-horizon operational training range.

HATT-X (High-speed Aeronautical Technologies Testbed)

HATT-X is a flying laboratory based on the ROBOT-X and used in weapon system evaluation trials. Current research topics include quaternion-based all-attitude AHRS (Attitude and Heading Reference System), second-generation autopilots, fibre-optic gyros and neural network implementations of selected autopilot functions. A turbine-powered version of the airframe is also being configured. HATT-X is being considered as a carrier for the testing of pyrophoric flares.

Badger

BADGER is a tele-operated 6-wheeled all terrain vehicle used as a tractor for towing surface targets to represent the hostile threat posed by an enemy tank. The vehicle is capable of 40+ km/h with excellent off-road, all-weather mobility, has a load capacity exceeding 300 kg, and incorporates fail-safe features. It is using an inexpensive forward-looking video camera, which relays the information to the target control centre. Use of the tunable video link allows the operation of several (up to 5) targets simultaneously. CF will be taking delivery of the first series of the target systems manufactured by Bristol Aerospace of Winnipeg, Manitoba this summer.

Barracuda

The BARRACUDA is a 7.3 m surface target boat system capable of speeds of up to 35 knots, remotely-controlled from more than 10 nautical miles. It can mimic high-speed naval tactics and a variety of operational scenarios. It can be equipped with visual, radar and laser signature enhancements to present a convincing likeness of a variety of threats. The target system is based on an relatively inexpensive rigid fiberglass hull with inflatable collars, which allows operations in severe weather up to sea state 3. Audio, video and telemetry give the operator feedback on speed, course and position. It is manufactured by Bristol Aerospace of Winnipeg, Man.

ANCAEUS

ANCAEUS is a semi-autonomous vehicular control system which can be applied to any land or sea surface vehicle. The system features navigation using GPS and onboard data communication with fibre-optics. ANCAEUS uses a highly modular design concept with commercial hardware installed in a specially designed enclosure to meet military level reliability considerations. ANCAEUS is currently implemented on a remotely-controlled tractor for towing land target systems, on a hazardous materials handling robot and on a tractor vehicle system for detecting buried mines in an operational scenario; the system is also slated for use in applications involving increasingly autonomous robotic vehicles.

Articulated Navigation Testbed (ANT)

Research into unmanned vehicle platforms has resulted in the ANT, a high mobility, multi-unit, articulated steer, unmanned vehicle demonstrator with locomotion provided by six leg-wheel combinations. Adopting a multi-unit architecture means that it may be configured as either a 4X4 or an 8X8, depending on specific mission requirements. Articulated steering creates inherent obstacle avoidance for trailing axles, facilitates retro-traverse in marginal terrain conditions, and benefits autonomous or teleoperation because steering control of any of these configuration is transparent to the user as well as the command and control system and comes by virtue of intelligent design of the mechanical configuration. The ANT is controlled by the ANCAEUS vehicle control system.

Other Research Activities in Robotic Vehicle Technologies

The DRES is also supporting R&D in areas of machine intelligence and robotic movement control. Work focuses on guidance and control of both the vehicle in its environment (navigation, path planning, obstacle detection and avoidance), and the control of instruments such an cameras and manipulators. Recently, work has been completed on software simulation with graphical animation of a mobile machine controlled by a recurrent neural network; this work, which may lead to adaptive obstacle avoidance and navigation, is expected to be implemented on a mobile robot. Other R&D in this area involves the use of binary neural network for use in the control of an active suspension system for an off-road vehicle. The design of a behaviour-based simulator has been completed and is being implemented. Work continues on the control of a mobile robot using virtual presence hardware.

DREP

Robotics-related activity at DREP has focused on the methodology for efficient examination and identification of structural integrity of high-performance aircraft[C1-C12]. Ultrasonic, thermographic, and X-ray inspection techniques for CF-18 were addressed. The thrust of DREP's research is to explore and develop robotic scanning methods that can be applied directly to the aircraft structure to facilitate accurate, consistent, automated, in-situ inspection. A number of related activities are pursued in support of the objective.

- The objective of Integrated Radiographic Imaging System (IRIS) activity is to incorporate various radiographic imaging tools into an integrated working environment based on a 6 degree-of-freedom gantry robot. A 486-based PC is used to control the scanning motion of a selected imaging tool.

- Automated inspection and repair system for composite/aluminum honeycomb or monolithic composite laminate aircraft components comprises of a nondestructive testing (NDT) system to locate and assess extend of the damage, a 3-D vision system to determine the physical dimensions of the damaged area, a sensor data fusion system combining all the relevant results from a variety of NDT techniques to formulate an accurate 3-D flaw model of the damaged area, a computer-aided manufacturing (CAM) system to fabricate the replacement parts, and a robot repair unit to install the replacement parts and repair the damaged area.
- Neural-Network approach is used for object orientation classification in a noisy video image.
- Robotic positioning system for real-time radiography is using an inverse kinematics algorithm, and then determines the necessary excitations to the joint drive motors based on the computed joint values.
- Robot modeling and simulation is used to test and evaluate the robotic scanning system, and to determine the feasibility of an inspection concept, before hardware implementations are considered.
- Through-transmission ultrasonic inspection of composite structures was developed for use as an end-effector on a PUMA 560 industrial robot with rectilinear scanning capability on flat composite panels.
- Intelligent robot system for ultrasonic inspection of composite aircraft components involves a three dimensional vision system, a PUMA 560 robot and an ultrasonic imaging unit to perform e.g. ultrasonic inspection of a CF-18 landing gear panel.
- Telerobotic manipulation system for CF-18 NDT was developed under a contract.
- Mobile platform and a couplant system for the RSI C-yoke have been tested in a laboratory environment on a full-scale CF-18 stabilator.

Other robotic activities at DREP include fibre optic cable laying with an autonomous underwater vehicle, and mine countermeasures. Specifically, research is conducted to improve the positioning accuracy of the sonar on the seabed and to improve sonar image quality. Development of automatic bottom following for the towfish using active towfish control or winch control systems is considered. Also, a preferability study for a Remotely Controlled Minehunting System (RCMHS) for the safe search of mines involves considerations in both, ROV applications, and sonar image processing.

DREV

A remotely-controlled tracked vehicle, called Remotely Operated Placement Equipment (ROPE)[D1] was developed at DREV for EOD applications.

PRECARN

PRECARN is a not-for-profit corporation which fosters precompetitive research into artificial intelligence and robotics or, more generally, into "intelligent systems". This is an industry-led and managed organization which coordinates collaborative activities between corporations, federal and provincial governments, and universities. The Department of National Defence has a status of an Associate Member in the organization which facilitates full access to the knowledge and technologies resulting from PRECARN's activities. Broader networking is also a major part of PRECARN's strategy; industry and government members are regularly invited to tour university laboratories, and, in turn, university members are encouraged to talk to industry representatives at PRECARN conferences and workshops. Primarily, however, PRECARN coordinates and supports at a multi-million dollar level a number of very significant robotic projects:

The Advanced Process Analysis and Control System (APACS) is a network of systems that uses simulation and expert system technology to advance the automation of process systems. The project will lead to the development of an expert system capable of monitoring and interpreting a continuous process in real-time, including an advanced simulation facility for the diagnosis of plant malfunctions and the prediction of plant behaviour.

The Intelligent Graphics Interface (IGI) project aims to provide more effective human-computer interfaces for complex computerized systems. The application of interest here is the operation and management of real-time supervisory control systems, such as those used in telecommunications network management, power distribution and industrial process control.

The Autonomous Robot for a Known Environment (ARK) project will produce a self-contained mobile robot with capabilities specifically designed for operation in an industrial setting - the large floor at Atomic Energy of Canada CANDU in Mississauga, Ontario. Using its onboard sensor system, consisting of sonars, cameras, a dead reckoning systems and a laser range finder, the robot will be able to find its own way from one place to another, with no help from a human operator. As it moves about, the robot will compare a partial internal representation of its environment with the images from its cameras to determine its position and to help it plan its movements. The ARK robot will avoid any objects in its path, whether or not the objects are stationary or moving, and whether or not the objects are part of the internal representation.

The Telerobotics Development System (TDS) aims to develop three telerobotic systems of different scale and precision: a microrobot with sufficient precision to manipulate a single cell, a macro-robot capable of working on overhead electrical transmission lines and a research robot, for medium-sized applications requiring high precision.

The Knowledge-Aided Design (KAD) aims to discover new ways of assisting engineers to use computers to develop new and improved designs. Its objective is to introduce appropriate hardware and software technologies into the engineering design process to improve the effectiveness and productivity of teams of creative and experienced engineers working on large cooperative projects.

The Mining Automation Plan (MAP) aims to advance mining into a process control-based industry using mobile robotics by combining evolutionary and revolutionary innovations.

The Control of Robots for Future Applications (CORFFA) aim to demonstrate improved control of flexible dynamics and contact motion and demonstrate the results using simulations of the Space Station Remote Manipulator System and a typical in-vessel manipulator for a tokamak fusion reactor.

PRECARN is also actively involved with Canadian universities through its management of the Institute for Robotics and Intelligent Systems (IRIS), one of the fifteen federal Networks of Centers of Excellence. It is a $23.8 million, four-year project involving over 120 researchers at 18 Canadian universities from coast to coast. The research program of IRIS includes 22 projects within three areas of investigation:

- computational perception,
- knowledge-based systems, and
- intelligent robotic devices.

Results from this program are feeding directly into PRECARN's projects in a number of cases. PRECARN and IRIS conferences are held annually.

Military Colleges

Research and development activities within the National Defence Canada are also pursued at three military colleges. The Royal Military College (RMC) in Kingston, Ont. is particularly active in the classical robotics domain[E1-E29]. Scientists in three departments are involved: Electrical and Computer Engineering, Mechanical Engineering and Engineering Management. Six different themes are pursued:

- Modeling and Control. Kinematic and dynamic models are under development for manipulators available in the laboratory. The reachable and dextrous workspace for each of the manipulators is being mapped, and coordination tasks identified. The problem of interaction of dissimilar manipulators is of particular interest. Stiffness effects in flexible manipulators are under investigation with the goal to verify experimentally the conditions under which the stiffness effect is important. To obtain the equations of motion for a flexible manipulator with the endpoint constrained to move on a surface, a general formulation for the Jacobian and its derivative was developed and implemented in a simulation program. Other studies in the area involve: (1) a feed-forward controller based on a rigid model with various corrections for flexibility, (2) nonlinear adaptive control for distributed parameter systems with chemical reactors and flexible manipulators as practical applications, and (3) control the endpoint of a robot with flexible links when the endpoint moves on a surface with some friction causing stiction.

- Mulitprocessing. is studied using a VME bus with four 68020 processor boards using MPX multiprocessor multi-tasking Real Time Operating System and a CASE tool which specifies robotics control algorithms as task precedence graphs. The CASE tool determines allocation of tasks, creates a load module and automatically generates the initialization and communication code.

- Multiple Robot Coordination. The control of heterogeneous manipulators in a closed-chain configuration represents a significant challenge due to the differing kinematic and dynamic characteristics of each robot. The development of rountines to allow a more advanced robot to emulate one of lesser capability is pursued. A multiple arm controller and inter-processor communications protocol for the VME bus have been designed on the multiprocessing capability. It is to ensure collision-free motion when the arms operate independently and it is to enable the arms to perform coordinated movement.

- Sensors and Sensor Integration. Sensory models for monocular camera, laser range scanners, tactile sensors, and ultrasound images have been implemented in the simulation environment. A set of sesor integration algorithms have been also developed to feed the robot control module.

- A six-joint experimental anthropomorphic robot for research on flexible manipulators in 3D space is under development. Interchangeable links will facilitate testing of different materials and link shapes.

- A variety of simulation tools are under development; e.g. simulation of any open-loop, rigid, n degree-of-freedom robots with elastic joints and flexible links, and graphic simulation to test coordination and collision-avoidance algorithms for a multiple robot arm workcell.

The small Department of Engineering at the Royal Military College (RRMC) in Victoria, B.C. also contributed to the robotics theme, specifically on underwater robotics and acoustic telemetry[E30-E32] as well as control of flexible manipulators[E33-E34].

Military Robotics Workshops

Military robotic applications workshops[7-9] are held biannualy in Canada since 1987. They provide a forum for a review of progress in this area.

Artificial Intelligence and Robotics Steering Group

At the conclusion of the Second Workshop on Military Robotics Applications[10] it was decided that a Steering Group would be formed to meet perceived deficiencies in the area of networking regarding long term future developments in Artificial Intelligence and Robotics for defence. The Chairman of the Steering Group was appointed by the Associate Deputy Minister (Materiel). The Steering Group assumed no mantle of authority but operates as a facilitator to allow for a free flow of information between Government Agencies, industry, academia and R&D institutes. It attempts to focus attention on specific areas of interest and recommends courses of action to their respective sponsoring organizations or agencies. It also oversees the main themes and location of Workshops and recommends host organizations.

International Activities

Canada was the host for a related NATO Long Term Scientific Study which included robotics as a technology for combat engineer operations[11]. Canada participated in the recently completed work of NATO Research Study Group (RSG-18) on Operator-Robot Interaction (ORI) issues and is involved in The Technical Cooperation Program (TTCP), Technical Panel WTP-6, which includes robotics in its theme.

Industrial Activities

Canadian industry has been a leader in robotics for unstructured environments for a significant period of time. However, this subject is suffiently extensive to constitute a separate paper.

Conclusion

Defence robotics R&D in Canada is progressing on a number of fronts in DND establishments and colleges, and through collaborative efforts with Precarn and Canadian industry. On this foundation, Canada is also a viable partner in international information exchanges in this multidisciplinary technology.

Acknowledgments

It is gratefully acknowledged that this paper has been prepared on the basis of materials provided by Dr. R.S. Thomas, DG R&D/CRAD; Mr. D. Wakefield, DRDL/CRAD; Dr. R. Suart and Mr. C. Laforce, DRES; Dr. E. Solem, DSEA/ORA/NDHQ; Mr. K. Yeung, DREP; Mr. B. Montminy, DREV; Prof. M. Farooq, RMC; Prof. J. Collins, RRMC; and Lt (Navy) D.A. Holland, OIC Manta.

References

General

1. MGen George MacFarlane, Keynote Address to the First Workshop on Military Robotic Applications, *Proc. of the DREP/RRMC Military Robotic Applications Workshop*, DREP Special Report 87-1, Victoria, B.C., Canada, 11-13 August 1987.
2. MGen P.E. Woods, Opening Keynote Address: The Military Perspective on Robotics, *Proc. of the Second Workshop on Military Robotic Applications*, DCIEM Special Report 90-1, pp.1-3, Kingston, Ont., Canada, 8-11 August 1989.
3. Col. C.J. Mialkowski, Opening Keynote Address, *Proc. 3-rd Conference on Military Robotic Applications*, DRES Special Publication 149, pp.1-3, Medicine Hat, Alberta, Canada, 9-12 Sept. 91.
4. E. Solem, "Robotics as a Strategic Technology; A Futures Perspective," *Proc. of the Second Workshop on Military Robotic Applications*, DCIEM Special Report 90-1, pp.117-133, Kingston, Ont., Canada, 8-11 August 1989.
5. E. Solem, "Social and Economical Factors of Robotic Systems," *Proc. of the DREP/RRMC Military Robotic Applications Workshop*, DREP Special Report 87-1, pp.83-95, Victoria, B.C., Canada, 11-13 August 1987.
6. *A Feasibility Study into the Application of Robotic Technology for Army Operational Use*, Report for DSS Contract No. 017SV.W8477-8-CB18/S, 1989.

7. *Proc. of the DREP/RRMC Military Robotic Applications Workshop*, DREP Special Report 87-1, Victoria, B.C., Canada, 11-13 August 1987.

8. *Proc. of the Second Workshop on Military Robotic Applications* , DCIEM Special Report 90-1, Kingston, Ont., Canada, 8-11 August 1989.

9. *Proc. 3-rd Conference on Military Robotic Applications*, DRES Special Publication 149, Medicine Hat, Alberta, Canada, 9-12 Sept. 91.

10. Plenary Panel Session: Open Discussion, *Proc. of the Second Workshop on Military Robotic Applications* , DCIEM Special Report 90-1, p.425-430, Kingston, Ont., Canada, 11-13 August 1987.

11. I.O. Moen, Study Director; *NATO Long Term Scientific Study, Applications of Emerging Technologies to Combat Engineer Operations*, NATO, AC/243(LTSS)D/37, 1989.

DCIEM

A1. J.J. Grodski, Robotics in Defence Environments, *Proc. DREP/RRMC Military Robotic Applications Workshop*, Victoria, B.C., 11-13 Aug 1987, pp.73-78.

A2. P. Milgram, D. Drascic and J.J. Grodski, Stereoscopic Video + Superimposed Computer Stereographics: Application in Teleoperation. J.J.Grodski and M. Farooq (Eds.), *Proc. Second Workshop on Military Robotic Applications*, Kingston, Ont., pp.20-25, North York, Ontario: DCIEM/RMC, 1990.

A3. P. Milgram, D. Drascic, and J.J. Grodski, A virtual stereographic pointer for a real three dimensional video world. A chapter in: D. Diaper, D. Gilmore, G. Cockton & B. Shackel (Ed's), *Human-Computer Interaction -- INTERACT '90*, pp.695-700, Elsevier Science, 1990.

A4. S.G. MacLean, M. Rioux, F. Blais, J.J. Grodski, M. Milgram, H.K.L. Pinkney, B.A. Aikenhead, Vision System Development in a Space Simulation Laboratory, *Proc. of SPIE - The International Society for Optical Engineering*, vol.1395, pt.1, pp.8-15, (Conference of the Intern. Soc. for Photogrammetry and Remote Sensing; Commission 5: Close Range Photogrammetry Meets Machine Vision); Zurich: SPIE, 1990.

A5. P. Milgram, D. Drascic, and J.J. Grodski, Enhancement of 3-D video displays by means of superimposed stereo-graphics, *Proc. 35th Annual Meeting of Human Factors Society*, pp.1457-1461. San Francisco, CA., 1991.

A6. J.J. Grodski, P. Milgram, D. Drascic, S. Zhai, "Real and Virtual World Stereoscopic Displays" , *Canadian Computer Show and Conference.*, Toronto, Nov 1992.

A7. D. Drascic, S. Zhai, P. Milgram, J.J. Grodski, "Augmenting Reality: Combining Stereoscopic Video with Stereoscopic Graphics", *Proceedings of 1992 International Conference on Three Dimensional Media Technologies (3DMT)*, Montreal, Nov. 1992.

A8. D. Drascic, "Stereoscopic Vision and Augmented Reality", *Scientific Computing & Automation*, 9(7), 31-34, Gordon Publications Inc, Morris Plains, New Jersey, June 1993.

A9. P. Milgram, D. Drascic, and J.J. Grodski. Stereoscopic Vision for Mobile Telerobotic Control, *US Patent #5,175,616* (1992).

A10. D. Drascic, J.J. Grodski, P. Milgram, P. Wong, K. Ruffo, and S. Zhai, ARGOS: A Display System for Augmented Reality, *Proc. Video programme and extended abstract, INTERCHI '93 Conference on Human Factors for Computer Systems*, Amsterdam, The Netherlands, 25-29 April 1993.

A11. J.J. Grodski, and P. Milgram, Real and virtual world stereoscopic displays for teleoperation, *Proc. NATO/DRG Seminar on Robotics in the Air-Land Battle*, pp.9-25. Paris 1991.

A12. D. Drascic, P. Milgram, and J.J. Grodski, Learning Effects in Telemanipulation with Monoscopic Versus Stereoscopic Remote Viewing, *Proc. IEEE, Systems, Man and Cybernetics Society Conference*, pp.1244-1249, Boston, MA., 1989.

A13. D. Drascic, M.A.Sc. Thesis: *An Investigation of Monoscopic and Stereoscopic Video for Teleoperation*, University of Toronto, 1991.

A14. D. Drascic, Skill acquisition and task performance in teleoperation using monoscopic and stereoscopic video remote viewing, *Proc. Human Factors Society 35th Annual Meeting*, pp.1367-1371, Sept. 2-6, 1991, San Francisco.

A15. S. Zhai and P. Milgram, A telerobotic virtual control system, *Proc. SPIE Vol. 1612, Intelligent Robotic Systems: Cooperative Intelligent Robotics in Space II*, Boston, 11-13 Nov. 1991.

A16. M. Krueger and P. Milgram, Issues in the use of interactive stereoscopic displays for exploring hazardous environments, *Proc. 11th Congress of International Ergonomics Assoc. (IEA)*, Paris, July, 1991.

A17. D. Drascic and P. Milgram, Positioning accuracy of a virtual stereoscopic pointer in a real stereoscopic video world, *Proc. SPIE Vol. 1457-38, Stereoscopic Displays and Applications II*, pp.1-12, San Jose, CA, Feb. 1991.

A18. M. Krüger, M.A.Sc. Thesis: *Adaptation in Depth Perception Using Stereoscopic TV Displays.*, University of Toronto, 1991.

A19. S. Zhai and P. Milgram, Human-Robot Synergism and Virtual Telerobotic Control, *Proc. Annual Meeting of Human Factors Assocn of Canada HFAC/ACE*, Hamilton, Ont. 25-29 Oct. 1992.

A20. K. Ruffo and P. Milgram, Effect of Stereoscopic + Stereovideo "tether" Enhancement for a Peg-in-Hole Task, *Proc. An. Conf. IEEE Systems, Man, and Cybernetics Soc.*, Chicago, 19-21 Oct., 1992.

A21. P. Milgram and M. Krueger. Adaptation effects in stereo due to on-line changes in camera configuration, *Proc. SPIE Vol. 1669-13, Stereoscopic Displays and Applications III*, San Jose, CA, 9-14 Feb. 1992.

A22. D.Drascic and J.J. Grodski, Using Stereoscopic Video for Defence Teleoperation, *SPIE Vol 1915 Stereoscopic Displays and Applications IV*, 1993, to be published.

A23. P. Milgram, S. Zhai, and D. Drascic, Applications of Augmented Reality for Human-Robot Communication, *IROS '93*, Yokahama, Japan, July 26-30, 1993.

A24. S. Zhai, P. Milgram, and D. Drascic, An Evaluation of Four 6 Degree-of-Freedom Input Techniques, *INTERCHI '93 Conference on Human Factors in Computer Systems*, Amsterdam, Apr. 1993.

A25. S. Zhai and P. Milgram, Human Performance Evaluation of Manipulation Schemes in Virtual Environments, *VRAIS '93 Virtual Reality Annual Symposium.*

A26. S. Zhai, "Investigation of Feel for 6DOF Inputs: Isometric and Elastic Rate Control for Manipulation in 3D Environments", *Proc. Human Factors Society and Ergonomics Society 37th Annual Meeting*, Seattle, 1993.

A27. S. Zhai, P. Milgram, "Human Performance Evaluation of Isometric and Elastic rate Controllers in a 6 dof Tracking Task", *Proc. SPIE Vol. 2057, Telemanipulator Technology*, Boston, Sept 1993.

A28. D.M. Stampe and J.J. Grodski, "Low Cost Software-Based Rendering and Stereoscopic Interfaces for Teleoperation and Virtual Reality," AGARD 76-th Symposium of Aerospace Medical Panel on Virtual Interfaces: Research and Applications, Lisbon, Oct 1993.

A29. J.J. Grodski and G. Immega, RObotic Muscle ACtuator (ROMAC) Theory, Proc. *DREP/RRMC Military Robotic Applications Workshop*, Victoria, B.C., 11-13 Aug 87, pp.41-48.

A30. G. Immega and J.J. Grodski, RObotic Muscle ACtuator (ROMAC) Robotic Applications, *Proc. DREP/RRMC Military Robotic Applications Workshop*, Victoria, B.C., 11-13 Aug 87, pp.49-54.

A31. J.J. Grodski and G.B. Immega, Myoelectric control of compliance on a ROMAC protoarm, *Proc. Int. Symposium Teleoperation and Control*, Bristol, U.K., July 88, pp.297-308.

A32. J.J. Grodski, G.B. Immega, and P.D. Lawrence, *Myoelectric Control of Actuators.* Canadian Patent 1,294,393, issued 21 Jan 92.

A33. J.J. Grodski, G.B. Immega, and P.D. Lawrence, *Myoelectric Control of Actuators.* U.S.Patent 4,964,061, issued 16 Oct 90.

A34. J.J. Grodski, G.B. Immega, and P.D. Lawrence, *Myoelectric Control of Actuators.* U.K.Patent, issued 91.

A35. D.M. Stampe, E. Reingold and J.J. Grodski, "Operator Gaze Position Control Interfaces: Investigation of Psychophysical and Operational Parameters," AGARD 76-th Symposium of Aerospace Medical Panel on Virtual Interfaces: Research and Applications, Lisbon, Oct 1993.

A36. J. Lipitkas, O. Bock, G..M.T. D'Eleuterio, and J.J. Grodski, Parametric Control of Point-to-Point Robotic Arm Movements using Artificial Neural Networks, *Proc. 7-th CASI Conference on Astronautics*, Ottawa, 4-6 Nov. 1992, publ. Canadian Aeronautics and Space Institute (CASI), pp.167-173.

A37. O. Bock, G.M.T.D'Eleuterio, J. Lipitkas, and J.J. Grodski, "Parametric motion control of robotic arms: A biologically based approach using neural networks," *Telematics and Informatics 10*, pp61-68, 1993.

A38. O. Bock, G.M.T.D'Eleuterio, J. Lipitkas, and J.J. Grodski, "Parametric motion control of robotic arms: A biologically based approach using neural networks," *Goddard Conf. Space Applic. Artif. Intelligence*, Greenbelt, USA, 1993.

A39. J.J. Grodski, D. Stampe and P.J. Jowett P.J., *Multiplexed Analog Control Link for an Explosive Ordnance Disposal Robot*, DCIEM Report No.91-65, 1991.

A40. J.J. Grodski, Demonstration of an Analog Communications Link for an Explosive Ordnance Disposal Robot, *Proc. ABCA-5 Conference*, Toronto, 1991.

A41. J.J. Grodski and D. Stampe, A Multiplexed Two-way Solid Communications Link for Teleoperated Robots, *Proc. 3-rd Conference on Military Robotic Applications*, Medicine Hat, Alberta, DRES Special Publication #149 (1992), pp.266-271.

DRES

B1. A.B. Markov and C.G. Coffey, A Rocket-Boosted Sea Launched Target System, *Third International Bristol RPV Conference*, Bristol, U.K., 13-15 September 1982.

B2. A.B. Markov, C.G. Coffey and T.E. Ollevier, DRES Rocket-Boosted Target Programs, *CASI Flight Test Symposium*, Medley, AB., 11-12 April 1984.

B3. S.G. Penzes, C.G. Coffey, H. Hudema, B. Boulter and A.B. Markov, ROBOT-X Recovery System Parachute Trials, *CASI Flight Test Symposium*, March 1985.

B4. A.B. Markov, C.G. Coffey, H.P. Hudema, D.N. Benson, S.G. Penzes, T.V. Meidinger and R.A. Eirich, ROBOT-X Proof-of-Concept Flight Trials: A System Review, *CASI Flight Test Symposium*, Montreal, 18-19 March 1987.

B5. E. Solowka, A. Markov, J. Bernacki, M. Hickey and D. Benson, Development of a Low Cost Autopilot for Aerial Target Systems, *CASI Flight Test Symposium*, Montreal, 18-19 March 1987.

B6. A.B. Markov and R.W. Herring, HATT-X: A High Performance Flying Laboratory for Autonomous Unmanned Aircraft, *Proceedings of the 3rd Conference on Military Robotic Applications*, Medicine Hat, AB., September 1991.

B7. A.B. Markov and R.W. Herring, HATT-X: A High Performance Flying Laboratory for Autonomous Unmanned Aircraft, *Proceedings of the 3rd Conference on Military Robotic Applications*, Medicine Hat, AB, September 1991.

B8. D.N. Benson, A.R. Burges, C.G. Coffey, R.W. Herring, H.P. Hudema, T. Meidinger, A.B. Markov and S.G. Penzes, Aerial Target Research of DRES: Rocket-Boosted Target Technologies and Beyond, *DRES 50th Anniversary Monograph*, in press.

B9. R.W. Herring, A.B. Markov et al., Aerial Target Research at DRES: Rocket-Boosted Target Technologies and Beyond, *DRES 50th Anniversary Monograph*, in press.

B10. A.B. Markov and R.W. Herring., HATT-X, A High Performance Flying Laboratory for Autonomous Unmanned Aircraft R&D", pp.136-142, *Proceedings of the 3rd Conference on Military Robotic Applications, Medicine Hat*, Alberta, 9-12 September 1991.

Mobility

B11. D.M. Hanna and T.F. Stupich, Mobility Performance Comparison of Robotic and Conventional Vehicles, *Proceedings DREP/RRMC Military Robotic Applications Workshop*, pp. 32-35, Victoria, BC, 11-13 August 1987.

B12. D.M. Hanna, R.J. Anderson and J.E. Tragenza, Terrain Previewing for an Active Suspension System, *Proceedings of the 3rd Conference on Military Robotic Applications*, pp 261-265,Medicine Hat, Alberta, 9-12 September 1991.

B13. S. Sankar, A. Dhir and V.S. Shankhla, Simulating Shock and Vibration Qualification of Unmanned Ground Vehicles Negotiating Off-Road Terrains, *Proceedings of the 3rd Conference on Military Robotic Applications*, pp 280-287, Medicine Hat, Alberta, 9-12 September 1991.

B14. A.W. McCormac, D.M. Hanna, and J.E. McFee, Unmanned Vehicle Mobility: Limits of Autonomous Navigation, *Proceedings of the 3rd Conference on Military Robotic Applications*, pp 301-310, Medicine Hat, Alberta, 9-12 September 1991.

Badger & Barracuda

B15. D.H. Saint and J.W. Funk, A Remotely Piloted Mobile Tank Target, *Proceedings of the DREP/RRMC Military Robotic Applications Workshop*, pp 153-159, Victoria, B.C., 11-13 August 1987.

B16. D.H. Saint, Field Tests and Future Development of the Tele-Operated Badger Robotic Vehicle, *Proceedings of the Second Workshop on Military Robotics Applications*, pp 256-262, Kingston, Ontario, 8-11 August 1989.

B17. J. Funk and G. Gillis, Design Summary and Potential Applications of a Naval Target Control System, *Proceedings of the 3rd Conference on Military Robotic Applications*, pp 288-295, Medicine Hat, Alberta, 9-12 September 1991.

Command/Control/Navigation

B18. M.G. Henders and R.H. Chesney, Ground Control Stations for RPVs - Present and Future, *Proceedings of the DREP/RRMC Military Robotic Applications Workshop*, pp 162-165, Victoria, B.C., 11-13 August 1987.

B19. R. Chesney, A Command and Control System for Autonomous Unmanned Air Vehicles, *Proceedings of the Second Workshop on Military Robotics Applications*, pp 242-247, Kingston, Ontario, 8-11 August 1989.

B20. D.J. Mackay, R.W. Toogood, and L. El-Rayyes, Symbolic Generation of the Equations of Motion for Flexible-Link Robot Arms, *Proceedings of the Second Workshop on Military Robotics Applications*, pp 326-333, Kingston, Ontario, 8-11 August 1989.

B21. D.R. Weiler, M.G. Henders, R.H. Chesney, and D. Ehman, Payload Centred Control for Unmanned Aircraf", pp 74-80, *Proceedings of the 3rd Conference on Military Robotic Applications*, Medicine Hat, Alberta, 9-12 September 1991.

B22. T.E. Ollevier, The Use of Low Cost Differential Mode GPS Receivers in Unmanned Vehicles, *Proceedings of the 3rd Conference on Military Robotic Applications*, pp 81-85, Medicine Hat, Alberta, 9-12 September 1991.

B23. V.C. Aitken and H.M. Schwartz, Motion and Structure from Planar Motion in Noisy Monocular Image Sequences, *Proceedings of the 3rd Conference on Military Robotic Applications*, pp 177-184, Medicine Hat, Alberta, 9-12 September 1991.

B24. S.A. Barton, Periodic and Chaotic Signal Generation in a Recursive Artificial Neural Network, *Proc. CRAD SIPWG Workshop on Neural Networks*, DREA, Dartmouth, N.S., May 1992.

B25. S.A. Barton, The Effect of Sensory Input on Trajectories Generated by Recurrent Neural Networks, *DRES SR-589*, January 1993.

B26. S.A. Barton, Two Dimensional Movement Controlled by a Chaotic Neural Network, *Proceedings of 1st IFAC International Workshop on Intelligent Autonomous Vehicles*, University of Southampton, UK, April 1993.

B27. R. Eirich and A. Kramer, A Generic Semi-Autonomous Ground Vehicle Control System, *Proceedings of the 3rd Conference on Military Robotic Applications*, pp 68-73, Medicine Hat, Alberta, 9-12 September 1991.

DREP

C1. K.K. Yeung, Colour Recognition Techniques to Detect Copper-based Particles in Filter Debris Samples and the Expert Diagnostic Chart for the Filter Debris Samples from the Main Gear Box of the SeaKing Helicopter, *Proceedings of the AMDU FDA Workshop*, AMDU, CFB Trenton, 9 Dec 1992

C2. K.K. Yeung, et al., A COMSCAN Investigation of Glass Reinforced Plastic/Foam Sandwich Material, *Proceedings of the Second International Conference on Sandwich Construction*, University of Florida, 9-12 Mar 1992.

C3. K.K. Yeung, P. Zakarauskas, A. McCray, A Neural Network Approach for Object Orientation Classification, *Proceedings of the SPIE*, Vol. 1569, *Conf. on Stochastic and Neural Methods in Signal Processing, Image Processing and Computer Vision*, San Diego CA, July 1991.

C4. K.K. Yeung, An Inverse Kinematic Algorithm for the DREP/Philips Real-time Radiographic Manipulator System, *Proc. 3-rd Conference on Military Robotic Applications*, DRES Special Publication 149, pp.129-135, Medicine Hat, Alberta, 9-12 Sept. 91.

C5. K.K. Yeung and A.G. McCray, Robot Simulation Software for the Design of a Thermographic Inspection System for the CF-18 Aircraft, *Proc. of the Second Workshop on Military Robotic Applications*, DCIEM Special Report 90-1, pp. 398-405, Kingston, Ontario, Canada,8-11 August 1989.

C6. K.K.Yeung, E.T Karkut, and T. Miller, An Intelligent Robot System for the Ultrasonic Inspection of Composite Aircraft Components, *Proc. of the DREP/RRMC Military Robotic Applications Workshop*, DREP Special Report 87-1, pp. 96-105, Victoria, B.C., Canada, 11-13 August 1987.

C7. G. Sedun, Application of Robotics to Aircraft Manitenance: A Case Study, *Proc. of the DREP/RRMC Military Robotic Applications Workshop*, DREP Special Report 87-1, pp.106-114, Victoria, B.C., Canada, 11-13 August 1987.

C8. K.K. Yeung and P.D. Lawrence, A 3-D Vision System Using Space encoded Spot Projections, *Proceedings of the SPIE Vol. 728*, 1986, pp.172-180.

C9. E. Jackson and J. Ferguson, Design of ARCS-Autonomous Remotely Controlled Submersible, *ROV 1984 Conference Proceedings*, Marine Technology Society, San Diego, California, pp.122-125.

C10. C. Thomas, Sea Trials and Operation of the ARCS Autonomous Submersible, *ROV 1985 Conference Proceedings*, Marine Technology Society, San Diego, California, pp.122-125.

C11. R. Thomas, Control Software in the ARCS Vehicle, *Proceedings of the Fourth International Symposium on Unmanned Untethered Submersible Technology*, 1985, Marine Systems Engineering Laboratory, University of New Hampshire, Durham, New Hampshire, pp.334-342.

C12. R. Thomas, The ARCS and DOLPHIN Untethered Remotely Operated Vehicles, *Oceanology 86 Proceedings*, Society for Underwater Technology, Brighton, U.K.,1986, pp.113-118.

DREV

D1. W. Slaght and W.J. Robertson, *Remotely Operated Placement Equipment (ROPE)*, DREV Report R-4494/88.

RMC

E1. K. Ziarati, J.-C. Piedboeuf, et R. Hurteau, Controleur anticipatif pour un robot ayant une articulation elastique et une membrare flexible, *14ieme Congres de mechanique apliquee CANCAM '93*,

E2. S.S. Lim and M. Farooq, Slew Manoeuvre Control for Large Flexible Spacecraft under Random Noise, *31st Conference on Decision and Control*, session TA-2, paper #3, pp. 1297-1302, Tucson, Arizona, Dec. 1992.

E3. J.G.A. Beaulieu, M. Farooq, T. Sheppard, and C. Wortley, The Allocation of Hard Real-Time Tasks to Heterogeneous Processors in Robotic and Digital Control Systems, *1992 International Conference on Industrial Electronics, Controls, Instrumentation, and Automation (IECON '92)*, pp. 571-576, San Diego, CA. , Nov. 1992.

E4. M.B. Broughton, The Use of a Novel Magnetoresistive Angle Sensor for Controlling the Speed of a Switched-Reluctance Motor, *Canadian Conference on Electrical and Computer Engineering*, Toronto, ON. , Sept. 1992.

E5. J.-C. Piedboeuf, R. Hurteau, et K. Ziarati, Logiciel de simulation de robots flexibles et de leurs controleurs, *Conference Canadiennes sur l'automatisation industrielle*, pp. 13.13-13.17, juin 1992.

E6. J.G.A. Beaulieu, M. Farooq, C.M. Wortley, and T. Sheppard, A Hard Real-Time Task Non-Preemptive Scheduler for Heterogeneous Processor Sets, *9th IEEE Workshop on Real-Time Operating Systems and Software*, Carnegie-Mellon University, Pittsburgh, PA., May 1992.

E7. S.S. Lim, M. Farooq, and N.U. Ahmed, Vibration Control for Large Flexible Spacecraft Subject to Slew Manoeuvering, *ISMM Conference on Computer Applications in Design, Simulation, and Analysis*, pp. 67-70, Orlando, Florida, Mar. 1992 (invited paper).

E8. S. Bruder, M. Farooq, M.M. Bayouni, Robotic Heterogeneous Multi-Sensor Fusion with Spatial and Temporal Alignment, *30th IEEE Conference on Decision and Control (CDC '91)*, Session W2-5, paper#3, pp. 506-511, Brighton, England, Dec. 1991.

E9. M. Farooq. Data Sensor Fusion Architectures, *DREO Workshop on the Naval Applications of Sensor Integration and Data Fusion*, Ottawa, ON., Nov. 1991.

E10. V.K. Banerjee, C.S. Earl, T.D. Looke, and M. Farooq, An Optimal Obstacle Avoidance Path Algorithm for the Excalibur Robot, *3rd Conference on Military Robotic Applications, Session on Guidance and Navigation*, pp. 93-98, Medicine Hat, AB., Sept. 1991.

E11. T.D. Looke, M.M. Bayoumi, and M. Farooq, The Response of Flexible Manipulators to a New Computer Torque Strategy: Simulation Results, *3rd Conference on Military Robotic Applications, Session on Robotic Arm Controls*, pp. 150-156, Medcine Hat, AB., Sept. 1991.

E12. M.J. McKay, M. Farooq, and C. Wortley, Improving the Execution Speed of Robot Control Computations: An Allocation Scheme for Heterogeneous Processes, *3rd Conference on Military Robotic Applications, Session on Supervisory Control*, pp. 249-254, Medcine Hat, AB., Sept. 1991.

E13. J.-C. Piedboeuf, D. Dochain, R. Hurteau, and K. Bemmeur, Optimal Control of the Top of a Flexible Arm, *Canadian Conference on Electrical and Computer Engineering*, pp. 73.2.1-73.2.1, Sept. 1991.

E14. J.-C. Piedboeuf and R.J. Ferguson, Simulation of Lightweight Robots and Their Controllers, *Third Workshop on Military Robotics Applications*, pp. 143-149, Defence Research Establishment Suffield, AB., Sept. 1991.

E15. S. Bruder and M. Farooq, A Comparative Study of Sensor Fusion Techniques, *NATO Defence Research Group (DRG) 30th Seminar on the Defence of Small Ships Against Missile Attacks*, Vol. 1, pp. 26.1 - 26.17, Ottawa, ON., May 1991.

E16.M.B. Broughton, A General State-Space Method for Modelling Piecewise Linear Systems, *34th Midwest Symposium on Circuits ans Systems*, Monterey, CA., May 1991.

E17. M.B. Broughton, Simulation of Piecewise Linear Ciruits in State Space, *34th Midwest Symposium on Circuits ans Systems*, Monterey, CA., May 1991.

E18. T.D. Looke, M.M. Bayoumi, and M.Farooq, Simulation of Computer Torque Controllers for Flexible Manipulators, *34th Midwest Symposuim on Circuits and Systems*, pp. 505-508, Monterey, CA, May 1991. (Invited Paper)

E19. M.J. McKay, M. Farooq, and C. Wortley, A Heterogenous Multiprocessor Programming Method with Application in Robotic Control Systems, *34th Midwest Symposuim on Circuits and Systems, Session FA-2*, paper #5, pp. 954-958,Monterey, CA,, May 1991.

E20. J.-C. Piedboeuf and R.Hurteau, Modelling and analysis of a two-degeree-of freedom robot with a flexible forearm, *Canadian Journal of Electrical and Computer Engineering*, vol. 16, no.4, pp. 127-134, 1991.

E21. M.B. Broughton, "A State-Space Method for Rapid Simulation of Piecewise Linear Systems," *9th Ship Control Systems Symposium*, Bethseda, Maryland, Sept. 1990. (Invited Paper)

E22. J.-C. Piedboeuf and R.Hurteau, Analysis and Simulation of a two-degeree-of freedom robot with a flexible forearm, *Canadian Conference on Electrical and Computer Engineering*, pp. 26.2.1-26.2.6, Sept. 1990.

E23. A.Rouhi, S.Monckton, and M.Farooq, Estimation of Circular Arcs Using an L1 Estimator in Combination with the Hough Transform, *343th Midwest Symposuim on Circuits and Systems*, Session WAR9, paper #5, pp. 1146-1149, University of Calgary, Calgary.AB, Aug. 1990.

E24. J.-C. Piedboeuf and R.Hurteau, The use of Kane's equations to write a continuous model of a flexible robot as a step toward a discrete model, *Dynamics of Flexible Structures in Space (C.L Kirk and J.L Junkins, editors)*, pp. 305-19, First International Conference, Cranfield, UK, Computational Mechanics Publications, Boston, May 1990.

E25. J.-C. Piedboeuf and R.Hurteau,, et. S. Sarrailion, Esa: logiciel d'analyse et de simulation d'un robot flexible a deux degres de liberte, *Proceedings of the 9th International Conference MICAD*, pp. 379-391, Feb. 1990.

E26. S.Bruder, M.Farooq, and M.M Bayoumi, Robot Control 'C' Library RCCL," *2nd Workshop on Military Robotic Applications, Session 11, paper #4*, pp. 392-397, Royal Military College of Canada, Kingston, ON, Aug. 1989.

E27. T.D. Looke, M.M. Bayoumi, and M.Farooq, The Responce of a One-Link Flexible Arm to Variable Structure Control Using Sliding Surfaces, *5th IFAC Symposium on Control of Distributer Parameter Systems: IFAC Series #2*, Perpignan, France, June 1989, pp. 333-337.

E28. T.D. Looke, M.M. Bayoumi, and M.Farooq, A Sliding Mode Controller in the Output Space, *IFAC Symposium on Nonlinear Control System Design*, Capri, Italy, June 1989, pp. 127-132.

E29. J.-C. Piedboeuf and R.Hurteau, A non-linear model of a flexible rotatin arm, *Jurnees de l' Optimisation* 1988.

RRMC

E30 J.S. Collins and C.J. Randell,"A method for tracking narrow beam acoustic telemetry links," *Fourth Int. Advanced Robotics Programme Workshop Proceedings*, 10 pg., Genova, November 1992.

E31 C.J. Randell and J.S. Collins, "Auto-alignment of acoustic transducers in a multipath environment," *IEEE Pacific Rim Conference on Communications, Computer and Signal processing*, pp.714-717, Victoria, May 1993.

E32 J.S. Collins and C.J. Randell, "An acoustically sensitive robotic wrist," *1993 IEEE Robotics and Automation Conference Video proceedings*, Piscataway, 1993.

E33 C.J. Damaren and I. Sharf, "Simulation of flexible manipulators with inertial and geometric nonlinearities," submitted to *ASME J. of Dynamical Systems, Measurement, and Control*, July 1993.

E34 C.J. Damaren, "Passivity analysis of multilink flexible space manipulators," submitted to *AIAA J. of Guidance, Control, and Dynamics*, July 1993.

90C231

Etude PROTECTRON[1] : Définition de Plates-Formes Robotisées Télé-Opérées Bas Coût pour le Champ de Bataille[2]

Aubert Carrel *, Jean-Pierre Pouplard *, Jean-Louis de Moulins **,
Annie Lelarge **, Michel Dégé ***, Gérard Falcou ***
* ITMI, 86/90 Rue Thiers, 92513 Boulogne-Billancourt Cedex (France)
** GIAT industries, 13, Route de la Minière, BP 1342, 78022 Versailles Cedex
*** ECA, BP 242, 83078 Toulon Cedex

#149C99

Résumé

Les objectifs de l'étude PROTECTRON étaient de définir des plates-formes robotisées télé-opérées bas coût pour le champ de bataille, à partir de systèmes robotisés existants et de technologies acquises, et de statuer sur la faisabilité et l'intérêt de l'introduction à court terme de la robotique de "bas de gamme" sur le champ de bataille.

Pour cela, l'étude recense les possibilités d'emploi opérationnel, pour des missions de combat dans la zone des contacts, de systèmes robotisés inspirés de plates-formes existantes, en visant des mises en œuvre à l'horizon 2000.

Pour les cinq systèmes robotisés conjuguant le meilleur intérêt opérationnel et une faisabilité technique et économique, l'étude présente un dossier de prédéfinition complet (architecture organique nouvelle), un dossier industriel (coût, échéancier de production) et militaire (concept d'emploi et conséquences sur l'organisation des unités).

Abstract

The aim of the PROTECTRON study was to decide, on the basis of existing mobile robots and of available technologies, on the feasibility and the interest of short term, low cost unmanned vehicles on the battlefield, and to define teleoperated robots for reconnaissance, observation and neutralization.

The project is built over a detailed and comprehensive state of the art of existing robotic platforms and of their constituting technologies. It identifies the various operational missions in the contact zone that could be efficiently performed by robotic systems directly inspired from existing platforms. The perspective is to have operational PROTECTRONs for the year 2000.

For each of the five robotic systems offering the best operational interest with an economic and technical feasibility, the study presents three reports : a technical one describing the architecture and the main sub-systems, an industrial one that gives costs evaluation and production planning and a military one which details the operational use and its consequences over the organization of the military units.

The five PROTECTRONs include four unmanned ground vehicles :

- *SERAC, an anti tank robotic platoon,*
- *GOR, a system specialized in forward observation and surveillance,*
- *ROC, a combat robot adapted to urban warfare,*
- *TOTE, a general purpose amphibious teleoperated vehicle,*

and one unmanned air vehicle :

- *ERA, a contrarotative platform used by ground units for local forward reconnaissance.*

These systems can be prototyped with today's technologies. They are small size vehicles (400 to 600 kg) with no or little decision making capabilities but adapted to intensive operations. They should be fielded at battalion level for a unit cost within the $300 000 to $400 000 range.

The introduction of PROTECTRONs in the force could bring numerous operational gains such as

---

[1] PROTECTRON = Plate-forme RObotisée Télé-opérée EConomique pour le Tir, la Reconnaissance, l'Observation, la Neutralisation…

[2] Cette étude a été financée par la DGA / SEFT (Délégation Générale à l'Armement / Section d'Etudes et Fabrication des Télécommunications).

enlargement of the controlled area, surprise, increase in the speed of the maneuver, better safety, important decrease in the workload and in the taskforce dedicated to the mission with a reduction of risk of casualties compared to classic operations. These advantages can be achieved in the majority of theatres and tactical situations.

Introduction

Objectifs de l'étude

La robotique militaire a deux motivations : réduire les risques croissants encourus par le combattant, et accroître l'efficacité opérationnelle des systèmes d'armes qui pourrait être limitée par la présence même de l'homme.

Plusieurs études technico-opérationnelles ont déjà été réalisées, au niveau OTAN [1], ou au niveau national, entre autres aux USA [2] et en France [3,4,5].

De nombreuses plates-formes robotisées existantes (civiles ou militaires, opérationnelles ou expérimentales) sont susceptibles de remplir, dans une version adaptée, des missions militaires. Certaines de ces réalisations permettent d'envisager aujourd'hui des plates-formes bas coût et réalisables à court terme.

Les objectifs de l'étude PROTECTRON sont :
- de définir de telles plates-formes (les PROTECTRONs) ;
- de statuer sur la faisabilité et l'intérêt de l'introduction de la robotique de "bas de gamme" sur le champ de bataille.

Organisation industrielle et étatique

L'étude s'est déroulée entre septembre 1991 et mai 1993, sous la supervision d'un "Polynôme de Pilotage" étatique constitué de représentants de la Délégation Générale à l'Armement et de l'Etat Major de l'Armée de Terre.

Elle a été menée par un Groupement de trois industriels : ITMI, pour ses compétences dans le domaine de la perception, et pour son implication dans de nombreux projets de robotique mobile ; GIAT Industries, pour sa connaissance approfondie de l'emploi opérationnel des systèmes d'armes terrestres ; ECA, pour sa maîtrise de l'intégration de systèmes automatisés en particulier les robots sous-marins autonomes de chasse aux mines.

Synthèse des apports de l'étude

Les cinq contributions principales de l'étude PROTECTRON sont :

- Un état de l'art structuré et la réalisation de la première base de données exhaustive et détaillée sur les systèmes robotisés existants et les technologies qui les constituent.
- Une analyse pragmatique des possibilités offertes à court terme et dans la zone des contacts par des concepts résultant d'une transposition des systèmes robotisés existants, illustrés par des missions types.
- La définition de cinq systèmes PROTECTRON, architectures organiques nouvelles correspondant à une évolution plus ou moins importante des plates-formes existantes, et résultant de la convergence des analyses opérationnelle et technique (en particulier, évaluation du cycle industriel à mettre en place pour la production des robots et estimation du coût de série).
- Des conclusions techniques (liste des technologies que l'on peut considérer comme acquises, et recommandations concernant les recherches et développements technologiques prioritaires).
- Un bilan des apports opérationnels des cinq PROTECTRONs (adéquation à la mission, définition d'un concept d'emploi et conséquences sur l'organisation des unités).

Ces cinq points sont détaillés dans les chapitres suivants.

Analyse structurée de l'existant

Le premier apport de PROTECTRON est un état de l'art, international et approfondi, sous forme d'une base de données exploitable sur Macintosh. Cette source d'informations, unique à notre connaissance, contient aujourd'hui 191 systèmes robotisés existants, décrits par 236 rubriques commerciales et techniques, classés en 24 familles et 43 sous familles. Elle offre la possibilité de tris, de consultations sélectives, d'ajouts et de modifications.

La base de données couvre aussi les technologies de la robotique mobile, regroupées en 34 domaines techniques, qui couvrent cinq grands sous-systèmes : la charge utile, la plate-forme avec sa mobilité et sa navigation, les liaisons entre le robot et l'opérateur, le Poste de Contrôle-Commande (PCC) avec ses interfaces, et enfin la protection de l'ensemble.

Analyse des possibilités offertes par les systèmes robotisés existants

La confrontation directe entre les besoins opérationnels dans la zone des contacts et 32 systèmes robotisés représentatifs a permis d'identifier les missions militaires qui pourraient être exécutées de façon avantageuse par des robots dérivés de ces systèmes existants.

Parmi ces 32 systèmes étudiés, la critique opérationnelle a été détaillée sur une sélection de 13 d'entre eux, pour lesquels des missions-types ont été élaborées (et constituent une source très représentative des applications potentielles de la robotique dans la zone des contacts).

Définition de cinq systèmes PROTECTRON

Travaux réalisés

Parmi les 13 systèmes robotisés pour lesquels a été faite une critique opérationnelle poussée, cinq ont été choisis, sur des critères d'intérêt opérationnel, de faisabilité technique et de coût : ce sont les PROTECTRONs.

La synthèse progressive ayant mené à leur élaboration est schématisée Fig. 1. Pour chacun, la démarche a été la suivante :

Dans un premier temps, une "Fiche de Besoins Mission" a été rédigée : elle présente en détail les performances requises pour remplir la mission-type.

Ensuite a été élaboré un dossier technique complet, décrivant l'architecture capable à la fois de remplir au mieux la mission-type et de respecter des contraintes de coût et de faisabilité technique à court terme. Cette architecture comprend la décomposition fonctionnelle et les choix organiques associés.

Enfin, le PROTECTRON a fait l'objet d'un dossier d'évaluation opérationnelle et économique. Le volet opérationnel comprend une évaluation "utilisateur" (sur des critères de performances) et une évaluation "décideur" (sur les avantages opérationnels attendus et la place du robot dans l'organisation de l'Armée de Terre). L'évaluation "payeur" précise le cycle industriel à mettre en place (ainsi que l'échéancier de financement associé) pour développer et produire les parcs de systèmes correspondant à la dotation prévisionnelle. Le coût unitaire de série est estimé.

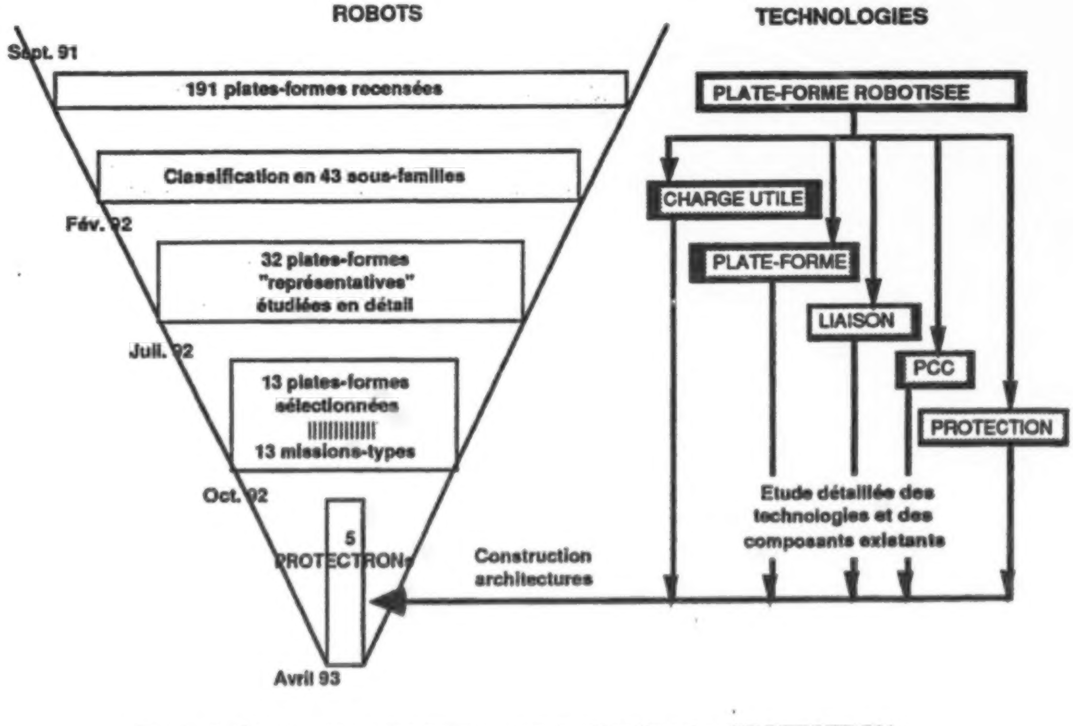

ROBOTS
TECHNOLOGIES
Sept. 91
191 plates-formes recensées
Classification en 43 sous-familles
Fév. 92
32 plates-formes "représentatives" étudiées en détail
Juil. 92
13 plates-formes sélectionnées
13 missions-types
Oct. 92
5 PROTECTRONs
Avril 93
Construction architectures
PLATE-FORME ROBOTISEE
CHARGE UTILE
PLATE-FORME
LIAISON
PCC
PROTECTION
Etude détaillée des technologies et des composants existants

Figure 1. Convergence des travaux vers la définition des PROTECTRONs

Tableau I : Les cinq PROTECTRONs

	SERAC	GOR	ROC	TOTE	ERA
Acronyme	SEction de Robots Anti-Chars	Groupe d'Observation Robotisé	RObot de Combat	Télé-Opéré Tout Emploi	Equipe de Reconnaissance Aérienne
Dérivé du robot : Fabriqué par :	Fire Ant Sandia Natl Labs	TMAP Grumman	TSR 700 "Wasp" IMI SIVAN	Rambler Kentree	Sprite ML Aviation SMR
Mission	A l'aide d'un ou de plusieurs robots, assurer et renouveler un guet anti-chars suivi d'un tir de destruction.	Surveiller une zone de façon autonome et collectivement, avec envoi d'alarme au poste de contrôle / commande.	Eclairer et couvrir la progression d'une unité d'infanterie en milieu urbain.	Reconnaître une coupure humide avant franchissement (entre autres).	Reconnaître une zone, par moyen aérien, en temps réel, pour le compte d'une unité mobile.
Description	Petite plate-forme 4x4 munie d'un système de détection automatique et de deux missiles anti-char Eryx[1].	Plate-forme identique à celle de SERAC, mais charge utile composée de capteurs de détection et d'identification.	Plate-forme 6x6 munie d'une charge utile d'observation et de tir, déployable en hauteur ou vers l'avant.	Plate-forme 6x6 amphibie (munie d'hélices) avec une charge utile composée de capteurs adaptés à la mission en cours.	Petit hélicoptère radio-commandé transmettant des images en temps réel.
Echelon d'emploi	Régiments de combat d'Infanterie et d'ABC[2].	Régiments de mêlée.	Compagnies de combat d'Infanterie et du Génie.	Unités élémentaires de combat des armes de mêlée, DLRG[3].	Régiment de mêlée et de reconnaissance.
Dotation	30 systèmes de 4 robots	16 systèmes de 5 robots (2 par Grande Unité)	150 robots (un par compagnie)	110 robots (2 par régiment)	50 systèmes (2 robots + 1 PCC)
Coût unitaire	1.8 MF	2.1 MF	1.8 MF	1.6 MF	2.1 MF

Les cinq PROTECTRONs

Le tableau I décrit rapidement les PROTECTRONs ; deux d'entre eux (GOR et ERA) sont présentés plus en détail en annexe, les trois autres (SERAC, ROC et TOTE) sont présentés Fig. 2 à 4.

Conclusions techniques

Position et domaines techniques couverts

Les cinq PROTECTRONs sont entièrement réalisables aujourd'hui, au niveau du prototype.

Ce sont des robots de petit gabarit (masse comprise entre 300 kg à vide et 600 kg en charge pour les PROTECTRONs terrestres, et inférieure à 50 kg pour l'aérien) disposant d'une bonne mobilité.

Cette première génération dispose d'une intelligence réduite ; ils n'ont donc pas (ou très peu) d'autonomie décisionnelle (toutes leurs actions sont contrôlées par l'homme). Trois d'entre eux disposent cependant d'automatismes élémentaires permettant soit une navigation semi-autonome (cas du robot aérien) soit une veille-détection automatique, robot à l'arrêt (cas de SERAC et GOR).

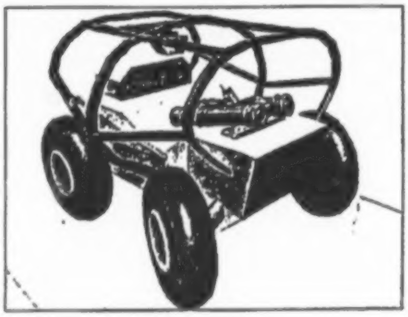


Figure 2 : Le PROTECTRON "SERAC"

---

[1] Eryx = missile anti-char filoguidé commercialisé par Aérospatiale.

[2] ABC = Armes Blindés Cavalerie.

[3] DLRG = Division de Liaison et de Reconnaissance du Génie.

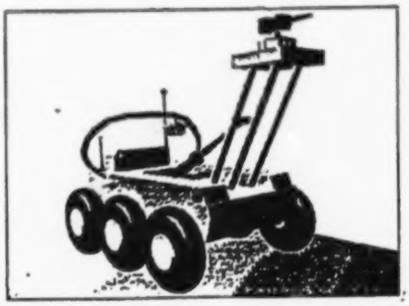


Figure 3: Le PROTECTRON "ROC"

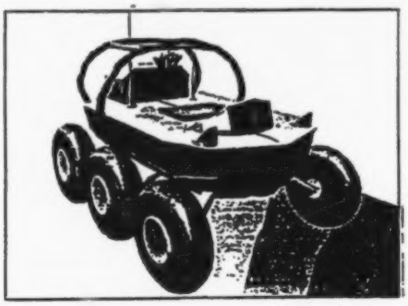


Figure 4: Le PROTECTRON "TOTE"

Ils peuvent partager certains sous-systèmes constituant ainsi des modules communs, tels que la plate-forme (leurs spécifications de mobilité et de franchissement étant très proches), les transmissions (par fibre optique avec radio de secours), le PCC (déportable depuis un véhicule "support" léger) ou encore la structure générale de contrôle/commande (télépilotage de bas niveau, avec retour vidéo). Seule la charge utile, éminemment variable selon les missions, est spécifique à chaque PROTECTRON.

Ils sont téléopérés à partir d'un Poste de Contrôle-Commande via une liaison moyenne portée (0.5 à 6 km pour les robots terrestres). L'opérateur assure tous les contrôles de pilotage (élaboration des itinéraires, exécution du transit) et de mission (identification, ordre de tir...). Cette configuration nécessite des liaisons robot->PCC large bande, robustes et fiables pour transmettre notamment le retour vidéo et une liaison PCC->robot numérique, sécurisée, pour transmettre les ordres.

Technologies des PROTECTRONs

L'étude détaillée des PROTECTRONs a fourni une synthèse des acquis techniques, c'est à dire des capacités qui résultent de l'intégration de technologies existantes. Ces points techniques, acquis et intégrés, couvrent tous les domaines de la figure 5 : Architecture générale - Capteurs pour la mission - Armes et outils - Déploiement de la charge utile - Mobilité de la plate-forme - Pilotage embarqué - Localisation - Protection et sécurité - Vision pour le pilotage - Liaison fibre optique et radio - Poste de Contrôle / Commande - Véhicule de servitude - Liaison avec l'échelon supérieur.

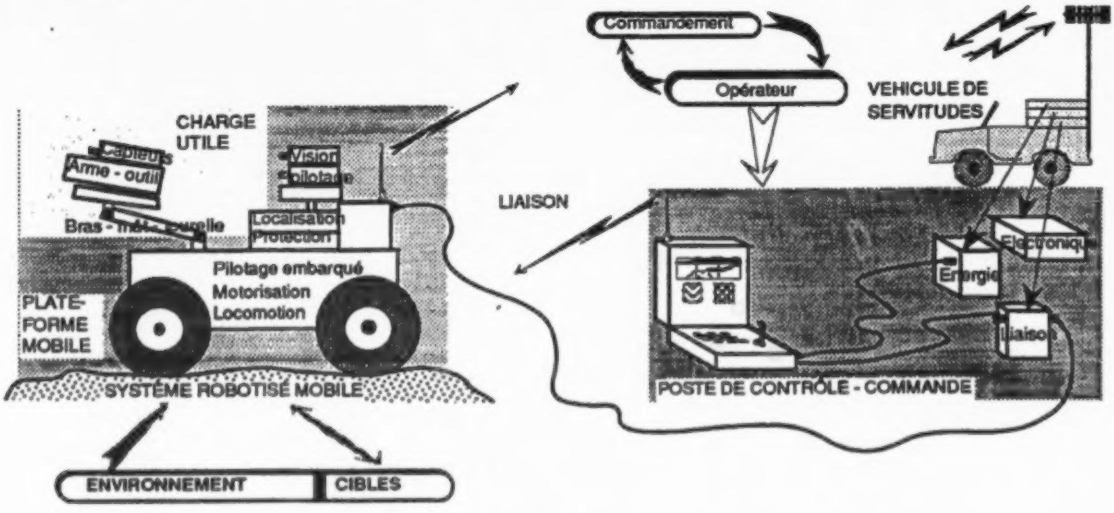

Commandement
Opérateur
VEHICULE DE SERVITUDES
CHARGE UTILE
Capteurs
Arme - outil
Vision pilotage
Bras - mât - tourelle
Localisation
Protection
LIAISON
Pilotage embarqué
Motorisation
Locomotion
PLATE-FORME MOBILE
Electroniq
Energie
Liaison
SYSTÈME ROBOTISÉ MOBILE
POSTE DE CONTRÔLE - COMMANDE
ENVIRONNEMENT
CIBLES

Figure 5 : Eclaté d'un PROTECTRON

Recommandations pour évaluer les performances de technologies liées aux architectures proposées

Les points suivants sont technologiquement acquis mais ils méritent une attention toute particulière, et même des expérimentations spécifiques, lors de la conception fine des PROTECTRONs :

- L'efficacité et la sécurité de la commande à distance des armes portées (missiles anti-chars Eryx, fusil mitrailleur 5.56 Minimi...) doivent être impérativement garanties pour assurer l'acceptabilité des PROTECTRONs de combat par les forces.
- La stabilité des petites plates-formes des quatre PROTECTRONs terrestres doit être étudiée par simulation de roulement en conditions extrêmes sur différents types de terrain.
- La robustesse de la fibre optique est très critiquée sans qu'il existe à notre connaissance d'évaluation opérationnelle conduite de façon scientifique. La fibre optique étant à court terme inévitable il apparaît indispensable de mener des expérimentations systématiques sur ce point.
- L'efficacité de la chaîne de vision tactique doit donner à l'opérateur une agilité d'observation et une capacité d'identification comparables à celles d'un homme en situation sur le terrain.
- L'efficacité de la chaîne de vision pour le pilotage à distance doit être optimisée pour la détection des zones carrossables y compris avec des caméras "bas niveaux de lumière".
- L'opérateur doit "comprendre" en permanence et sans effort d'interprétation la position de la plate-forme localement (par rapport à un objet proche) et globalement (dans une grande zone).

Recherche et développement à moyen terme

Certaines technologies, dont l'intégration pourrait apporter dans le futur une amélioration importante de l'efficacité opérationnelle des robots, ont été volontairement exclues des architectures proposées par manque de maturité. Nous recommandons vivement de soutenir un effort de Recherche et Développement dans ces domaines :

- Les automates et les assistances à l'opérateur permettant l'identification des cibles ou des contacts, comme les systèmes IFF (Identification Friend / Foe).
- Les techniques visant à améliorer les performances de mobilité à vitesse soutenue sur route, chemin ou tout terrain, grâce à des automates de pilotage (assistance à la stabilité de la plate-forme, commandes réflexes sécuritaires).
- La compression d'images pour la téléconduite en utilisant les algorithmes de compression classiques, et aussi des algorithmes prenant en compte les spécificités du mouvement de l'image en téléconduite.
- Les systèmes autonomes de navigation permettant à la fois d'acquérir une indépendance vis à vis des liaisons PCC-robot, d'accroître la rapidité de réaction et de réduire le nombre d'opérateurs. Cette autonomie peut être atteinte par des sauts technologiques successifs : navigation en tout terrain ouvert, suivi de route, suivi de chemin et navigation en tout terrain accidenté.

Bilan opérationnel des cinq PROTECTRONs

Utilisation intensive

Les cinq PROTECTRONs se prêtent tous à une utilisation intensive : quelle que soit la mission, ils peuvent être en action 20 heures sur 24 environ.

Echelon d'emploi : le Régiment

L'échelon "Régiment" semble le plus à même d'assurer le plein emploi de tous les systèmes envisagés. L'échelon de la Grande Unité est plus économique en nombre de systèmes nécessaires mais trop éloigné de la zone de combat pour apprécier les urgences et opportunités des actions à conduire au profit des unités élémentaires. A l'inverse, l'unité élémentaire n'est pas en permanence en situation d'utiliser un système robotisé.

Elargissement de l'espace de manœuvre

Tous les PROTECTRONs contribuent à donner au chef tactique un espace de manœuvre élargi : la frange du terrain qu'il fait parcourir à ses robots et dont il contrôle, grâce à eux, les accès, que ce soit par l'observation ou par le feu. Elle constitue pour lui un espace de sécurité supplémentaire, d'une largeur de 1 km à plus de 20 km selon les PROTECTRONs.

Effet de surprise et durée

Chaque PROTECTRON, quelle que soit sa mission, bénéficie d'un facteur "surprise", élément de son efficacité. La surprise est un produit de la discrétion (furtivité), de la rapidité et de la durée : seul un robot peut assurer pendant une dizaine d'heures une veille parfaitement immobile et silencieuse.

Accélération du rythme de la manœuvre

Les PROTECTRONs obtiennent l'effet recherché plus vite ou aussi vite qu'une unité conventionnelle à pied ou motorisée ; ils ne freinent pas le rythme de la manœuvre mais permettent au contraire de l'accélérer grâce à la précision des renseignements qu'ils donnent. Dès la mise en place effectuée, les renseignements arrivent en temps réel sous forme d'images.

Sûreté des dispositifs et précision des interventions

Par le renseignement de combat ou technique, ou encore par le tir, les PROTECTRONs participent à la sûreté rapprochée ou éloignée des dispositifs arrêtés ou en progression, en donnant un préavis sur l'arrivée de l'ennemi ou en donnant sa localisation et sa nature. Ces informations, capitales pour décider de la manœuvre à choisir, sont obtenues sans engagement d'unité d'homme alors qu'il faut généralement consentir à combattre pour les obtenir.

- Allègement des charges humaines

Tous les PROTECTRONs permettent à la fois de projeter une capacité dans une zone de danger, d'économiser les moyens humains, et d'éviter les pertes humaines.

Une mission effectuée par un PROTECTRON comparée à la même mission réalisée par une unité conventionnelle se situe, en matière d'effectif concerné, dans un rapport de 1 à 10, parfois plus.

L'effectif du système robotisé n'est jamais dans une situation de risque alors que l'unité conventionnelle l'est toujours. Un système robotisé permet d'économiser les unités en les réservant pour les phases du combat où elles sont irremplaçables. Enfin l'utilisation des PROTECTRONs aurait un effet sécurisant pour l'ensemble des unités concernées et pour le chef responsable.

Polyvalence de théâtres

Le domaine fixé comme cadre pour l'étude se limitait aux théâtres Centre-Europe et désert chaud. Les PROTECTRONs sont opérationnellement valables sur les deux théâtres, ainsi que sur tout autre à l'exception du théâtre tropical et des forêts denses.

Polyvalence de conflits

Les missions de robots envisagées dans l'étude sont, parfois au prix de quelques adaptations, adaptables à toutes les formes de crises, de conflits ou d'engagements actuellement admis comme plausibles.

Quelle que soit l'intensité des conflits, que les crises portent sur le maintien ou le rétablissement de la paix, les PROTECTRONs peuvent éviter d'engager des hommes dans certaines actions dangereuses, améliorer la sécurité des dispositifs, faciliter la prise de décision tactique.

Polyvalence des attitudes opérationnelles

Les PROTECTRONs conviennent aux besoins opérationnels, que la mission générale soit défensive, offensive, ou se réduise à un contrôle de zone.

Conclusion

Parmi les cinq PROTECTRONs définis on trouve quatre systèmes terrestres, et un système aérien. Ce sont des systèmes entièrement réalisables aujourd'hui, au niveau du prototype. Ils sont de petit gabarit, se prêtent à une utilisation intensive et n'ont pas (ou très peu) d'autonomie décisionnelle. Leur dotation se situe généralement au niveau régimentaire, et leur coût série est voisin de 2 MF.

Leur introduction dans les forces apporterait de nombreux avantages opérationnels, tels que l'élargissement de l'espace de manœuvre, l'effet de surprise, l'accélération du rythme de la manœuvre, une plus grande sûreté des dispositifs, un allégement des charges et des risques de pertes humaines, et ceci dans la majorité des théâtres et des situations tactiques.

Références

1. M. Bresson et al., "Robots in the Battlefield", Long Term Scientific Study, NATO Defense Research Group, Report AC/243-D/1024 & D/34 (NATO Restricted), July 1985.
2. D.R. Brown et al. (SRI International), "R&D Plan for Army Applications of AI and Robotics", Final Report ETL-0296, May 1982.
3. J.P. Pouplard, R. Mur et al. (AERO), "Programmation de Robots Mobiles Urbains - Analyse Opérationnelle", Rapport Intermédiaire Marché DRET n°84/455, Novembre 1985.
4. A. Briand, R. Baton, S. Muller (CEA / GIAT Industries), "Etude de Concepts de Robots de Combat", NATO DRG Seminar on Battlefield Robotics, ENSTA, Paris, 6-8 Mars 1991.
5. P.Raynal et al. (AERO), "Etude du Compromis Autonomie-Téléopération en Robotique Mobile Militaire", Rapport Technique n°2 Marché DRET n°91/418, Novembre 1992.

Annexe A : Présentation du PROTECTRON "GOR" ("Groupe d'Observation Robotisé")

GOR est une plate-forme robotisée téléopérée, qui a pour mission le guet, l'alarme et l'identification.

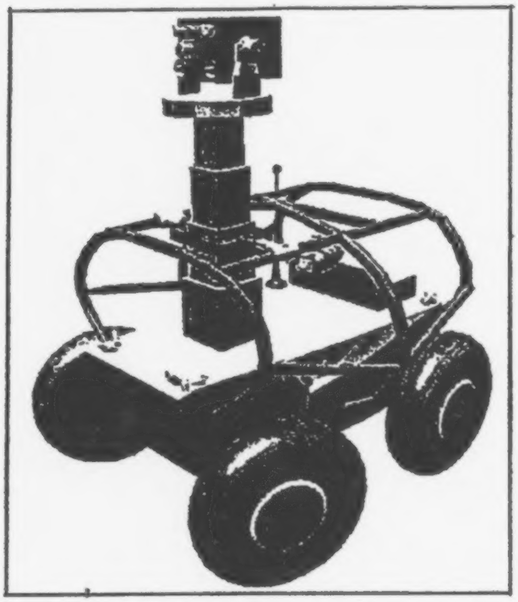


Figure A.1 : Vue en perspective de GOR

Descriptif

GOR est un outil télé-opéré d'observation éloignée, travaillant seul ou à plusieurs, de jour et de nuit. Basé sur un châssis de petites dimensions, très furtif, il peut se déplacer sur route, chemin ou tout-terrain facile. Son moteur diesel et son système de locomotion (4x4 à 2 roues directrices, garde au sol importante, suspension étudiée) lui donnent une bonne mobilité et de bonnes capacités de franchissement.

Afin de pouvoir détecter et identifier des véhicules ou des fantassins, il est muni de capteurs montés sur un mât télescopique, déployable à l'arrêt.

La détection se fait par l'association de capteurs acoustiques (microphones) et visuels (caméras jour et nuit). L'électronique de traitement est embarquée, la détection se fait donc de manière automatique avec envoi d'alarme à l'opérateur. En option, on peut monter une caméra infra-rouge ou un radar de type RB12B[1].

L'identification est faite par l'opérateur, au moyen d'une deuxième batterie de caméras avec zoom, orientable séparément.

La liaison entre le robot et le PCC transite par une fibre optique légère consommable. Une liaison de secours par radio permet de continuer la mission et de récupérer le robot en cas de rupture de la fibre (une antenne est alors déployée au PCC).

GOR est télé-opéré depuis une console portable, qui dispose d'interfaces ergonomiques pour la commande (joystick) et le contrôle : 5 écrans LCD vidéo couleur permettent de visualiser l'environnement du robot, sa position sur une carte ainsi qu'un tableau de bord pour aider au pilotage.

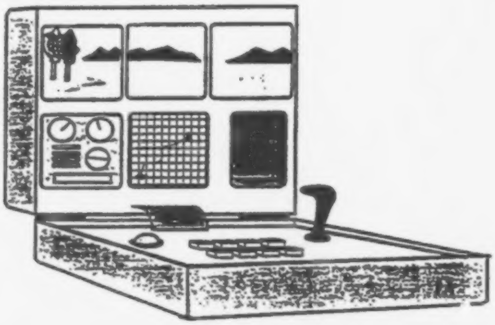


Figure A.2 : Console de Contrôle/Commande

Caractéristiques

- Masse : 450 kg
- Dimensions (Lxlxh, mât replié) : 2 x 1.2 x 1.5 m
- Vitesse max (en télépilotage) : 20 à 30 km/h
- Hauteur du mât / sol : 2.5 m (course 1.4 m)
- Secteur surveillé : gisement 60°, réglable +/- 90°
  site 20°, réglable +/- 30°.
- Performances en détection : char à 2000 m
  camion à 1000 m
  piéton à 200 m
- Prix série : environ 2.1 MF (selon configurations)

Mission

Le chef de mission choisit le nombre de robots qu'il veut déployer et leur emplacement respectif pour couvrir au mieux la zone d'intérêt. Chaque robot est alors télé-piloté par son opérateur jusqu'à son point de station (qui peut être distant de 4 km à vol d'oiseau) grâce au retour visuel des caméras de pilotage et au suivi de la position du robot sur une carte numérisée.

---

[1] RB12B = radar portable de surveillance du champ de bataille commercialisé par Thomson-CSF.

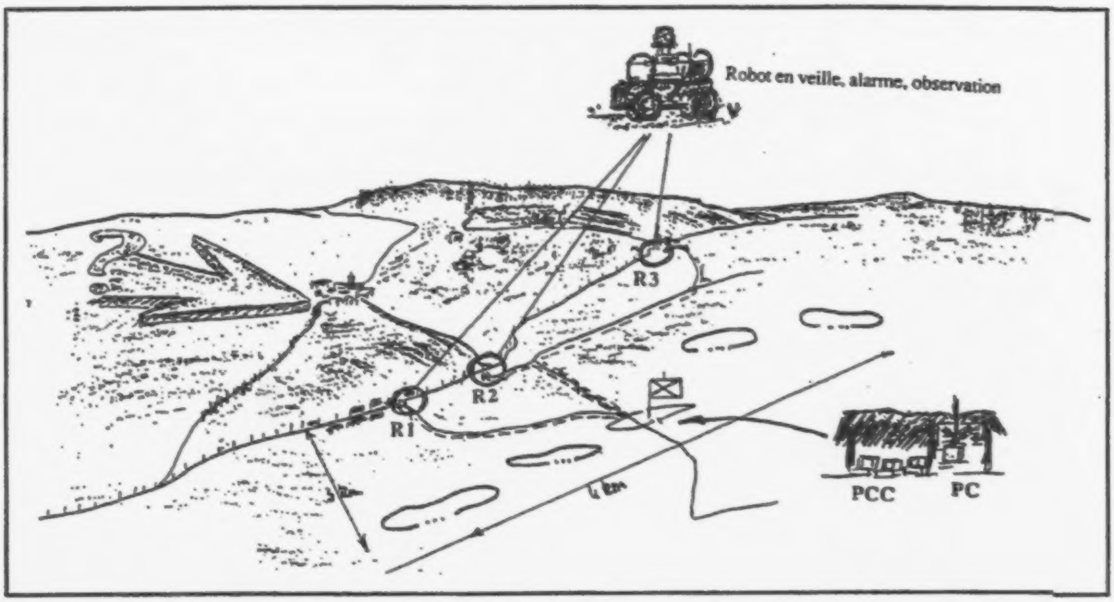

Robot en veille, alarme, observation
R3
R2
R1
6 km
PCC
PC

Figure A.3 : Mission-type du PROTECTRON GOR

Chaque GOR est mis à poste, son mât éventuellement déployé, et ses capteurs de détection orientés dans la direction convenable. Les robots sont alors prêts pour une veille silencieuse de plusieurs heures (9 au plus), et couvrent ainsi un front de 5 à 6 km de large. Un seul opérateur suffit pour cette phase de surveillance.

Toute détection déclenche une alarme. L'opérateur effectue alors l'identification de l'objet détecté grâce aux caméras du robot concerné, qu'il règle en orientation et en grossissement. Il informe l'échelon de commandement des objets cibles détectés (identification, localisation, déplacement).

A la fin de la mission, pour le retour, chaque robot est repris par son opérateur.

Avantages opérationnels

- Economie de personnel : l'équipe GOR comporte 9 hommes (contre 30 pour une section RASIT[1] par exemple).
- Aucun risque de perte (observateurs abrités dans les lignes amies).
- Grande discrétion lors de la mise en place, la veille et le repli.
- Continuité (une nouvelle mission peut commencer 1 h 30 après la précédente).
- Intérêt pour toute intervention et sur tout théâtre, à la fois pour la sûreté des dispositifs et le contrôle de zone.

---

[1] RASIT = radar de surveillance du champ de bataille commercialisé par Thomson-CSF.

Annexe B : Présentation du PROTECTRON "ERA" ("Equipe de Reconnaissance Aérienne")

ERA est une plate-forme robotisée téléopérée aérienne, qui a pour mission la reconnaissance à moyenne portée par la troisième dimension.

Descriptif

ERA est un hélicoptère de très petite taille. Il se déplace, sans axe d'avance privilégié, grâce à deux rotors coaxiaux contra-rotatifs entraînés par un moteur de 8 kW. Il est capable d'effectuer un vol stationnaire.

Il emporte un module d'observation constitué d'une caméra CCD haute résolution et d'une caméra à intensification de lumière, réglables en site et en zoom. Le réglage en gisement est obtenu par l'orientation de l'engin lui-même.

Les images filmées par ERA sont transmises en temps réel au PCC via une liaison hertzienne dans la bande 1-3 GHz. Le PCC est embarqué dans un véhicule léger, qui peut être mobile pendant la mission. Deux opérateurs assurent le bon déroulement de cette mission :

Le pilote dispose des interfaces de commande nécessaires au pilotage de l'engin lorsque celui-ci est en mode "télé-pilotage manuel" (consignes de cap, d'altitude...). Il visualise les images transmises ainsi qu'un tableau de bord l'informant sur l'état de ERA (attitude, niveaux...).

L'observateur commande l'orientation et le réglage des capteurs. Il peut aussi (mode "automatique") définir sur une carte numérisée les points de passage successifs du robot, qui sont alors suivis automatiquement (navigation basée sur un GPS). L'observateur visualise le retour vidéo "en direct" et en différé (les images transmises sont stockées sur magnétoscope) ainsi que la carte numérisée de la région, sur laquelle est incrustée la position du robot ainsi que le champ couvert par les caméras.

Caractéristiques

- Masse 50 kg
- Dimensions : diamètre rotors 1.6 m
  diamètre corps 0.6 m
  hauteur 1 m
- Vitesse max 130 km/h (vent 30 km/h)
- Portée 20 km
- Altitude max 2500 m
- Autonomie de vol : 2 heures
- Prix série : environ 2.1 MF (selon configurations)

Mission

ERA doit reconnaître un axe de pénétration avant le passage d'un convoi. Après son déchargement, le pilote le fait décoller en le télé-commandant à vue directe. ERA passe ensuite en mode semi- automatique et exécute les consignes de vol qui lui sont téléchargées depuis le calculateur du PCC.

Il se rend ainsi jusqu'au premier point d'observation, où le pilote en reprend le contrôle afin d'assurer un vol stationnaire sans dérive. L'observateur scrute la portion de terrain voulue, commande l'organe d'observation à son gré, demande éventuellement au pilote de modifier la position ou l'attitude du robot, et peut transmettre en radiophonie le résultat de ses observations au Poste de Commandement de l'échelon supérieur.

La distance typique d'observation est de 1000 m. Elle garantit l'acquisition d'une image très détaillée. A cette distance, ERA reste discret et est invulnérable face aux systèmes anti-avion et anti-missile.

Ensuite les consignes d'un transit vers le nouveau point d'observation sont téléchargées au robot.

La mission peut requérir que le PCC se déplace, soit pour maintenir l'intervisibilité avec ERA, soit pour des raisons tactiques. Ce déplacement peut se faire soit pendant une séquence de transit programmé du robot, soit pendant que celui-ci est en vol stationnaire "d'attente".

Avantages opérationnels

- Agrandissement de l'espace de manœuvre : ERA va chercher le renseignement jusqu'à 20 km du dispositif ami.
- Gain de temps : ERA se déplace à grande vitesse, en s'affranchissant des obstacles et des voies de communication.
- Localisation de l'ennemi avant le contact.
- Economie de personnel voire d'unités entières pour un même effet de sûreté et de renseignement, sans perte.
- Adaptabilité à tout type d'engagement.

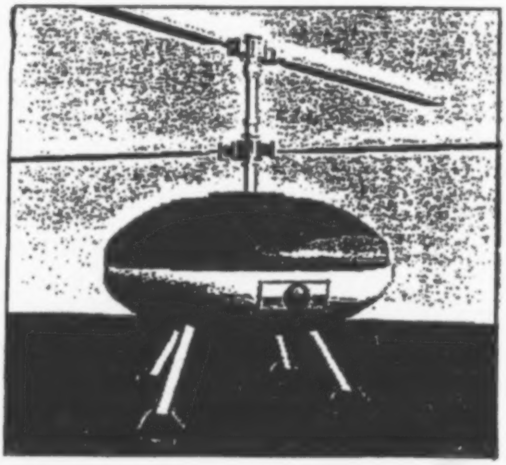


Figure B.1 : Vue en perspective de ERA

---

1 GPS = Global Positioning System.

904957

An Analysis Tool for the Backbone of the Automated Defence Data Network

Karen Bryden[1] & Dwaine Wattie[2]

149700

Abstract

An expert system is described which monitors a DND communications network, displays the status of the network and provides the best available advice on recovery and repair. The expert system represents a practical attempt to integrate an object-oriented relational database and an object-oriented graphical programming environment into a hybrid knowledge-based system. In this paper, the requirement and design of the expert system are discussed. As well, the integration of the database and knowledge base is discussed in terms of its motive, the potential benefits of such integration, the method in which it was accomplished, and finally the limitations of the method in achieving the desired benefits.

Introduction

The Automated Defence Data Network (ADDN) is a far flung message switching system which links elements of the Canadian Forces throughout Canada. Maintaining and operating this network poses a formidable logistics task for a number of reasons. Its sheer size and complexity requires a large well trained work-force. Built in the early part of the 1980's using then mature technology, it is not capable of being retrofitted to embed automated fault location and system restoral. Considerable expertise is required to operate the network and interpret system problems.

The Network Operations Advisor (NOA) is the second of two knowledge-based systems designed and deployed in the ADDN. Its job is to monitor the network, identify and log events affecting service and provide troubleshooting information to the operator. The scope of NOA encompasses the network at large and provides real-time data on the status of every major component. A future enhancement will allow commands to be sent directly to system computers to configure the network or perform diagnostics under operator control.

A high performance Graphical User Interface (GUI) was identified as a key requirement of NOA. In order to reduce the effort required for graphics development, a commercial object-oriented graphical programming environment called SuperCard® was adopted for the task. SuperCard® objects contain scripts which can be used effectively as rules, however the environment does not support frame-like knowledge representation. Representation of knowledge as frames was accomplished by adding an object-oriented relational database and merging the two into a hybrid knowledge- based system. The resulting system can be queried on-line by an external agent and its elements can be extracted for reuse in other systems.

The use of object oriented programming environments such as SuperCard® for construction of expert systems is well known [1]. The potential of modern relational database tools for representing domain knowledge has been investigated [2,3]. A rule base, stored as elements of a relational database, makes available the power of the relational data model for rule maintenance and browsing [4]. In addition, the coupling of database technology and knowledge-base technology is one avenue of current research on knowledge reuse and knowledge sharing [5].

In the following sections, the DND requirement is described, and the features of this requirement which had a major impact on the design of NOA are discussed. Following this, the operation of NOA is described briefly with an example. The integration of database and knowledge base within NOA is then discussed in terms of its motive, the potential benefits of such integration, the method in which it was accomplished, and finally the limitations of the method in achieving the desired benefits.

Problem Description

A system was required to oversee the operation of a wide variety of interactive subsystems which comprise the ADDN (Fig. 1). NOA would identify and provide advice on equipment and telecommunications problems, recognize network

1. Communications Research Centre (CRC), 3701 Carling Ave, Ottawa, Ontario K2H 8S2
2. Department of National Defence (DND)/Directorate of Information Systems Engineering and Management (DISEM 6-2-3), 257 Slater St, Ottawa, Ontario K1P 5H9

jeopardy situations and furnish recovery and repair procedures. It would additionally provide general background knowledge and information collection procedures. The end users of the system were to be network System Control Centre operators and node personnel. It was to be fielded on personal computers.

Because the ADDN carries classified traffic, a direct link between NOA and the ADDN was initially ruled out. However the benefit of having NOA's database automatically updated coupled with the operational advantage of using the GUI as a status-board was so compelling that a link was later added. However, the capability was left for data to be entered manually by the user.

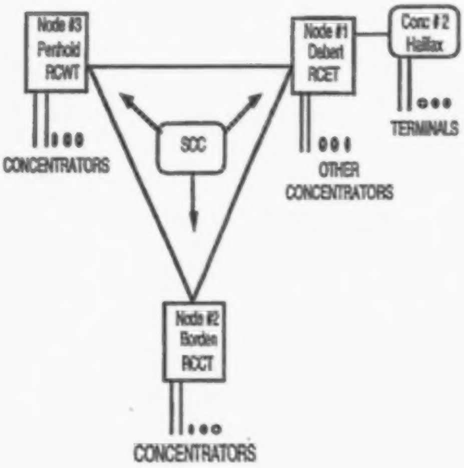


Figure 1 Automated Defence Data Network

Design Considerations

Initial experiments with a commercial expert system shell were not encouraging. The shell chosen was very slow on the Macintosh® host computer and a large development effort on the GUI appeared necessary. To avoid this, a combination of SuperCard® and HyperHIT-R[1]® was chosen, leading to the internal architecture of NOA shown in Fig. 2. SuperCard® is an object oriented programming tool with a graphical environment and superior programmability of colors of objects. HyperHIT-R® is a full-featured suite of relational database tools which can be called from SuperCard® scripts.

SuperCard®[2] scripts were found to be very capable as a medium for rules. Instantiation and unification (combinations of descriptive variables) are readily implemented, as are forward and backward chaining. Manipulation of the GUI is accomplished by SuperCard® scripts which respond to network events and actions of the user.

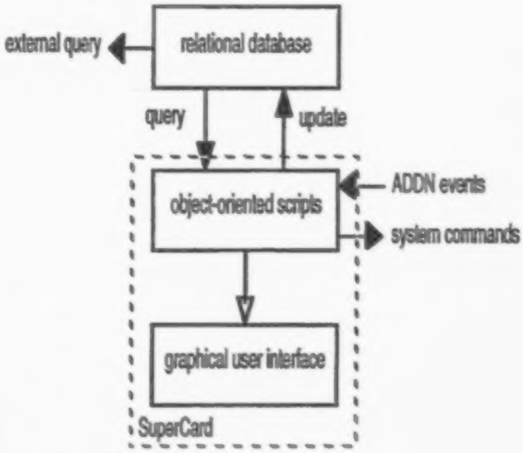


Figure 2, NOA Internal Organization

Object-Oriented Database

A relational database was used to record factual information in the database, including descriptions of problems and procedures, historical information on network outages and user notes. The database also represents aspects of the knowledge which cannot be readily represented in SuperCard®. Objects were created in the database representing the various subsystems and components of the ADDN, making it possible to use database queries in the reasoning process. Such an approach allows inheritance of properties, values and methods from classes to instances, and relations between objects to be explicitly represented.

In other efforts at integrating rule systems and database systems, the rules reside in the database. However this requires the domain knowledge to be encoded in the database query language, and severely limits the type of user interface which can be developed. In NOA, the database is copied into the user interface. This makes it possible to attach SuperCard® scripts to the objects and use the power of SuperCard® scripts and messages in order to reason on system events as well as operate the GUI. At system start-up, the GUI is rebuilt from the database, a necessary step to allow off-line maintenance of the database.

---

[1] Supercard is a trademark of Silicon Beach Software Incorporated, Aldus Corporation.

[2] Hyperhit is a trademark of Griffin Software.

A separate SuperCard® database tool was created for browsing and editing NOA's underlying structure. It can be used for example to change the number or identity of terminals attached to the ADDN or examine the conclusions reached during testing of the expert system. With the multi-user version of the database, it is possible to open and query the database while NOA is operational.

Object-Oriented GUI

A GUI was required which would provide multiple views of the network at various levels of detail, and allow unconstrained browsing. The ADDN has a great deal of built-in redundancy. Communications paths can be reconfigured in a myriad of different ways to overcome an outage. A graphical representation of the network was therefore required which would accommodate all possible system topologies without being overly complicated.

This was solved by physically moving the objects around on the GUI like chess pieces. Objects which can be logically relocated such as concentrators and terminals are represented as list items in the GUI and are associated with corresponding records in the database. The use of list items and icons to identify and manipulate objects in the database is an essential feature of NOA.

Maintenance of Context

During ADDN operation, system tables are duplicated at each of the three nodes to identify the current topology of the network and thereby the routing required for individual messages. Configuration messages are sent among the nodes at regular intervals and also when necessitated by network events. These messages are intercepted and sent to the NOA host computer, where they are decomposed into one or more records, structured reports which drive the GUI.

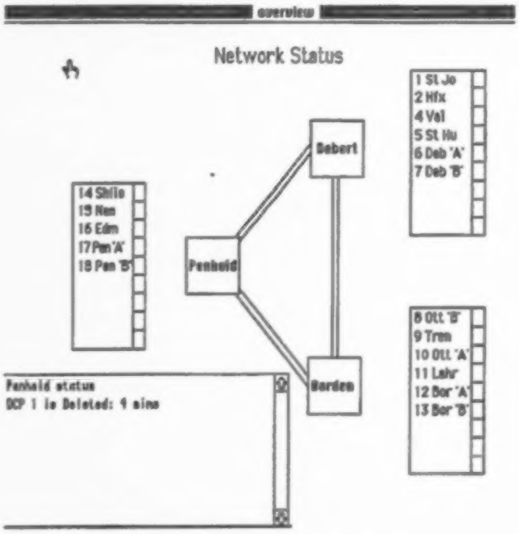
During periods when configuration messages are arriving rapidly, it is possible for messages to be received out of order, conflict with each other, or alter the context of the user's current investigation. Because the GUI is shared by the network and the user, messages could potentially generate erroneous status, change the current context or focus on another aspect of the system. The result could be very confusing to the user. Therefore each message received must be checked for validity and conflict with current status. When a message updates the GUI, the users context must be saved and restored.

Real-Time Reception of Events from the ADDN

The link to the ADDN was written in C language and makes use of SuperCard's callable interface. It receives messages from the network on the Macintosh® serial port, searches for key words and assembles formatted records for each object affected. SuperCard® polls the record buffer, reads the records and processes them.

Expert System Operation

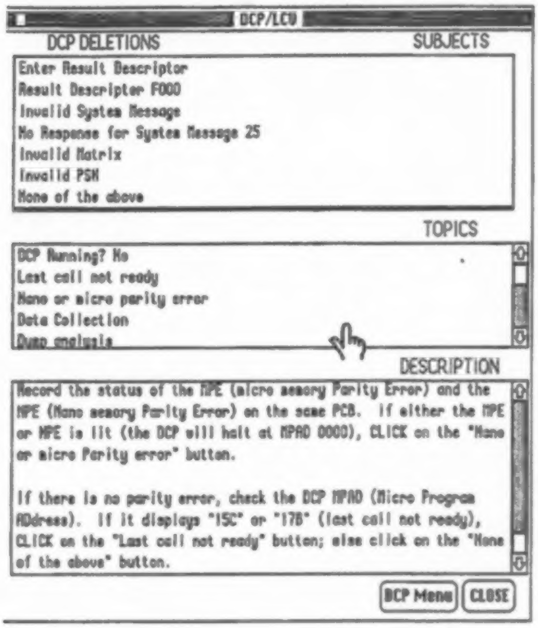
The top level of the GUI is shown in Fig. 3. From this view, the operator can examine the status of any node or concentrator in the ADDN. In this case, the operator will have noticed that the Penhold icon is yellow (indicating jeopardy) and clicking on the icon, will discover that DCP 1 (Data Communications Processor #1) has been deleted for 4 minutes. Using the pop-up menu at Penhold, the operator zooms in to obtain an internal view of the node. DCP 1 appears in red in this view and, clicking on the DCP 1 icon, the operator will observe that the result descriptor reported by the system is F000.



Penhold status
DCP 1 is Deleted: 4 mins

Figure 3, Overview of ADDN

Using the pop-up menu on DCP 1 to investigate DCP deletions, the diagnostic window shown in Fig. 4 will overlay the node view, giving an explanation of that result descriptor. By clicking on the topics, the operator can then browse at random through the best available advice on troubleshooting the problem. For example, clicking on "DCP Running?" and volunteering "No" to the question which appears will cause the answer to appear beside the question and provide a description of further tests. At any time, the operator can reconsider an answer, pursue another line of reasoning or defer the investigation while doing something else.



Enter Result Descriptor
Result Descriptor F000
Invalid System Message
No Response for System Message 25
Invalid Matrix
Invalid PSN
None of the above

DCP Running? No
Last call not ready
Nano or micro parity error
Data Collection
Dump analysis

Record the status of the MPE (micro memory Parity Error) and the NPE (Nano memory Parity Error) on the same PCB. If either the MPE or NPE is lit (the DCP will halt at MPAD 0000), CLICK on the "Nano or micro Parity error" button.

If there is no parity error, check the DCP MPAD (Micro Program ADdress). If it displays "15C" or "17B" (last call not ready), CLICK on the "Last call not ready" button; else click on the "None of the above" button.

Figure 4, Analysis of DCP Deletion

Knowledge Management and Reuse

The domain of NOA encompasses that of the first knowledge-based system developed for the ADDN, the Concentrator Maintenance Advisor (CMA). The CMA was written in M1®. One task in developing NOA was to extract as much of the CMA knowledge base as possible, for reuse in the larger system. Only a modest part of the information could be extracted, working on the knowledge base with a text editor. Such experiences are all too common, and have led to efforts to create standards in the expert systems field.

There are several motives behind standardization efforts [9]. Among them, research will advance more quickly if expert systems can be built on top of existing knowledge bases. Knowledge could be shared among cooperating systems. Generic knowledge bases may form the basis of reusable knowledge libraries. Companies are seeking to manage their knowledge assets not only due to the high cost of their development but also the commercial advantage they confer.

Several approaches to the problem of standards are being pursued [6]. Among those under study are standards for knowledge representation such as the Initiative for Managing Knowledge Assets (IMKA) and methods for translating between knowledge representations such as the Knowledge Interchange Format (KIF). Studies are also in progress on run-time interfaces between object oriented databases and other intelligent agents, and interfaces between relational databases and knowledge representation schemes [7].

An attempt was made with NOA to integrate a relational database, a knowledge representation scheme using rules and frames and an object-oriented graphical user environment into a single functional unit. The advantage to be gained from this is the ability to use both knowledge base and database tools for problem solving. However a suitable external interface to the knowledge base offers possibilities for knowledge sharing and knowledge reuse (in the absence of current standards) without having to impose a fixed knowledge representation scheme.

In order to reap these benefits, the integration must take place in such a way that (a) the knowledge base elements are represented in the database and (b), the paths are reciprocal (eg. changes made to the database will be known to the inference engine). Furthermore, a database server must be developed which is capable of interpreting the internal representation of the knowledge. For example in [4], a knowledge dictionary was created which allows the power of the relational database model to be applied in the analysis of relationships among rules. In [8], the need for further research on rule "compilers" is shown as well as the need for more research on tools for supporting knowledge base queries.

Modern database tools make it possible to embed domain knowledge in the database. This was not considered a viable alternative for NOA, particularly due to limitations in the user environments. However in NOA, a practical degree of integration was achieved by duplicating objects in the database within the SuperCard® environment. The disadvantage of this approach is that in addition to increasing processing overhead, the inference engine is not notified of changes made to the database from the external interface. This limits the potential for an on-line database interface to a read-only status (although the limitation is not serious in this application).

Results and Summary

An expert system has been written which reads ADDN system event messages, displays the status of the ADDN in a multi-level GUI and provides a wide variety of information to the user, including the best available advice on recovery and repair. The GUI is completely non-modal, allowing the user to browse at random.

In the design of NOA, a relational database, a knowledge representation scheme and a graphical user environment have been integrated into a single functional unit with some limitations. This scheme provides a powerful development environment, and through the means of database query, allows much of the internal structure of NOA to be made available to an external agent. This capability can be used to allow NOA to provide read-only information to another system on-line, and allows the structures of NOA to be extracted for use elsewhere.

A primitive knowledge base browser built as an aid for knowledge base development allows the knowledge engineer to trace the inferencing process through the database by manual navigation. The object hierarchy as well as the current state and history of events of the ADDN can be extracted. A database server is needed to identify the links between the rules (which are partly explicit and partly built into the SuperCard® message hierarchy), interpret the linkages to objects, and explain what has happened as the result of an event. It remains to be investigated how to accomplish this task.

References

1. "expert system optimizes engine efficiency", The Motor Ship, June 1990.
2. R.E. Leroux and J.D. Wilson, "Data-Driven Expert System for PC Board Diagnostics", Proc. 3rd Symposium/Workshop on Applications of Expert Systems in DND, May 1991.
3. K.A. Sosin, E Marois and P.R. Roberge, "A Knowledge Based System for Battery Selection in Avionics", Proc. 3rd Symposium/Workshop on Applications of Expert Systems in DND, May 1991.
4. B. Jansen and P. Compton, "The Knowledge Dictionary: A Relational Tool for the Maintenance of Expert Systems", Proc. Intl. Conf. on Fifth Generation Computer Systems", Nov. 1988.
5. R.M. MacGregor, "Knowledge-Representation Standards Effort", 7th IEEE Conference on Artificial Intelligence Applications, Feb. 1991.
6. R. Neches et al, "Enabling Technology for Knowledge Sharing", AI Magazine, Vol 12 No 3, Fall 1991.
7. D. McKay, T. Finin and A. O'Hare, "The Intelligent Database Interface", Proc. Eighth National Conf. on Artificial Intelligence, 1990.
8. M. Stonebraker, "The Integration of Rule Systems and Database Systems", IEEE Trans Knowledge and Data Engineering, Vol 4 No 5, Oct. 1992.
9. C. Hall, "Why standards? The CAIA '91 Workshop on Standards for Artificial Intelligence", IEEE Expert, Dec. 91.

404945

A Distributed Coordination Model for Battle Staff Tasks

Luc Lamontagne
Defence Research Establishment Valcartier
2459 Blvd. Pie XI north
P.O. Box 8800
Courcelette, Québec, Canada
G0A 1R0

#149701

Abstract

This paper presents a framework for the distributed coordination of individual tasks of Battle Staff members. The application problem, identified following a preliminary analysis of checklists used in the Canadian NORAD Region Operations Control Center, is described. A preliminary model for the coordination of partially ordered and related tasks executed by a group of battle staff members is developed. This intuitive distributed coordination model leads to a series of activities alternating between selection of intended tasks by an individual and acceptance of this task by the group. Agreement is reached when coordinating agents accept that the established individual sequencing of the intended tasks represent a compromise between individual preferences without violating group constraints.

1. Introduction

This work is directed toward the understanding and design of decision aids for the coordination of battle management tasks executed in organizations such as NORAD command posts. At the present time, very few tasks have been automated and most are still being executed by the Battle Staff individuals. Battle staff members exchange information, conduct individual activities and cooperate on common tasks to accomplish their common mission. Decision aids designed for such organizations should not only provide individual services to their main users but also help them manage their interactions with each other. The principal aim of this study is to illustrate that coherence and improvement of group performance can be reached by adapting individual procedures/activities using distributed coordination techniques.

In this paper, we present an intuitive model for battle staff group coordination. The model to be investigated should ideally be incorporated in distributed decision aids to coordinate BS members activities in command posts where a functional hierarchy, the members roles, and the individual procedures are already established. Decision aids including such a coordination scheme may act as agendas recommending sequenced tasks (actions) to be undertaken by members. Such systems may easily be combined with a graphical representation of military checklists.

The military application problem motivating this study of group coordination is described and the distributed decision aids environment is formulated in section 2. In section 3, a description of the coordination model is presented in two parts: the first part illustrates deliberations leading to hypothetical task selection, and the second part is centered around the interactions between coordinators to reach an agreement. Section 4 is a discussion of further aspects related to group coordination. Due to the classified nature of NORAD operations, the content of this paper makes no direct reference to the ROCC Battle Staff procedures.

2. Problem description

2.1 The military application

While conducting activities that are part of a battle management process, each member of a Battle Staff is assigned a series of tasks to execute. These tasks come from the series of responsibilities describing the roles members play in the organization hierarchy of the Battle Staff. Tasks are divided in two types: context dependent tasks and persistent tasks.

This arbitrary division is deemed necessary to represent the various roles of Battle Staff members.

A checklist may be considered as a partially ordered procedure to be followed by an officer under a change of alert status. It is used as a memo to keep track of the implementation of battle management operation plans. A subset of the tasks assigned to an officer may be found as action items on individual checklists. We refer to these tasks as context dependent since their execution is subject to their relative position in the checklist and depends on the checklist current status implementation of other members of the organization.

The remaining tasks, which we call persistent, are not included within checklists. We make the distinction between context-dependent and persistent tasks mainly because some BS members do not make use of checklists. Among the persistent tasks, some are executed on a periodical basis. For example, the Weather Officer is responsible for assessing almost every hour the meteorological categories of various sites. Some other persistent tasks are executed upon request from other members of the battle staff. One interesting property of the persistent tasks is that their execution normally takes place more than once in the battle management process; the number of times they are selected varies from one situation to another.

Both context-dependent and persistent tasks might necessitate direct interaction between Battle Staff members. However it is important to note that their assignment to individuals is static. In the NORAD context, BS members have dissimilar expertise and a series of dedicated checklists. In general, a task can only be executed by a single officer. This is not always the case for every military organization; for example, in Rescue Coordination Centers (RCC), checklists are associated with individual cases rather than with individuals.

The overall list of tasks to be executed over a period of time by a battle staff member is the aggregation of the context-dependent (action items) and persistent tasks. In figure 1 relationships describing high-level interactions between agent tasks are illustrated in hierarchical manner to represent the organizational structure of the Battle Staff. For the purpose of our study, we assume that individual task flows are centered around the execution of the checklists action items, with possible inclusion of the persistent tasks.

Due to the nondeterministic nature of the battle management environment, the order of task execution is expected to be kept dynamic to cope with varying conditions or periods of increased tension. This assumption results from a preliminary analysis of the checklists. The order in which the action items are presented on paper does not necessarily represent a strict order of execution and hence indicates that a partial temporal order between tasks should be considered.

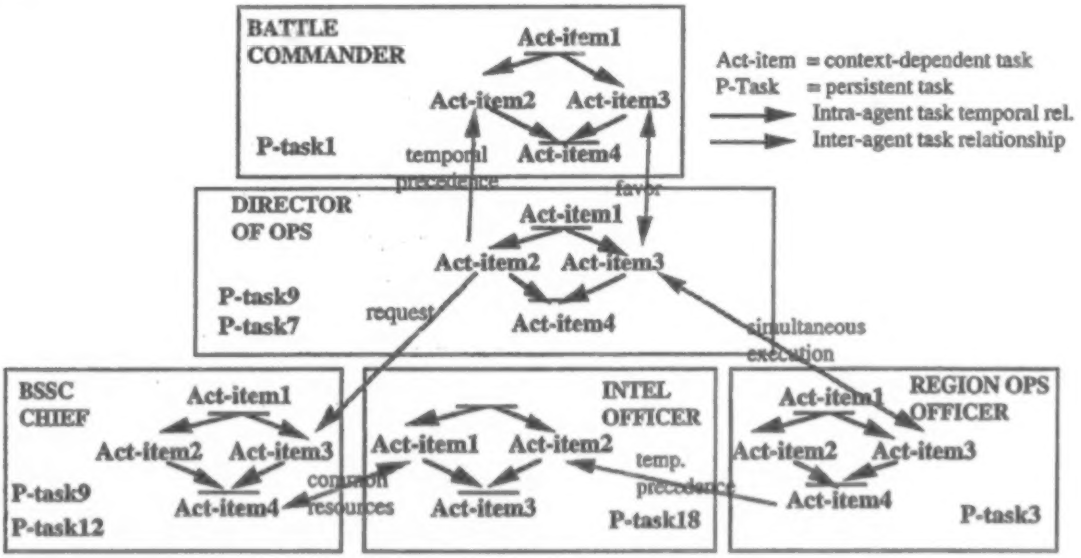

BATTLE COMMANDER
Act-item1
Act-item2
Act-item3
Act-item4
P-task1
Act-item = context-dependent task
P-Task = persistent task
Intra-agent task temporal rel.
Inter-agent task relationship
temporal precedence
favor
DIRECTOR OF OPS
Act-item1
Act-item2
Act-item3
Act-item4
P-task9
P-task7
request
simultaneous execution
BSSC CHIEF
Act-item1
Act-item2
Act-item3
Act-item4
P-task9
P-task12
common resources
INTEL OFFICER
Act-item1
Act-item2
Act-item3
P-task18
temp. precedence
REGION OPS OFFICER
Act-item1
Act-item2
Act-item3
Act-item4
P-task3

Figure 1 : Relationship Graph of Battle Staff Tasks

Moreover, comparison between various members checklists reveals many analogies, cross-reference or other relationships between tasks and even similar task patterns. It is expected that modification or shuffling of tasks during the course of action will take place to accomodate other members expectations, avoid creating conflicts between procedures, and improve group performance.

2.2 The distributed framework

Decision aids built for organizations where humans exchange information, conduct individual activities and cooperate on common tasks to accomplish their common mission pose some problems [4] [8]. Too often, automation efforts are limited to providing tools in response to individual data manipulation needs, thereby ignoring an integrative group approach. Battle staff members are still left with the responsibility to execute additional tasks, to interact with other BS members as well as with the decision aid system.

In this paper, BS members are mapped in a one-to-one fashion with agents responsible for coordinating their activities with other members activities (figure 2). Coordination of group tasks and of checklists action items is accomplished through exchanges among users and decision aids.

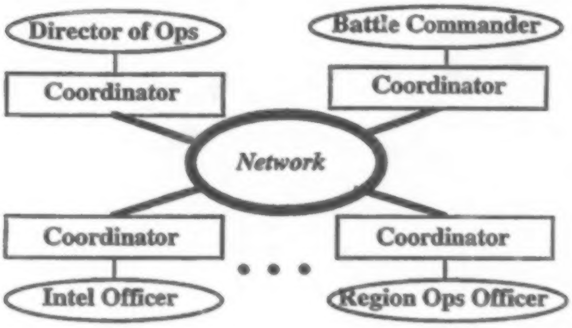


Figure 2: Individual Decision Aids for Group Coordination

A distributed environment is appropriate because of the distributed nature of the battle staff activities, the dissimilar expertise of the BS members, and the functional hierarchical structure of the Battle Staff organization [6]. Furthermore, as partial automation of the command post activities progresses in the future, the work load associated with task execution will be increasingly distributed between the decision aids and their users. Centralized management of hundreds of procedures including dozens of high-level tasks, which can be divided into many subtasks, would make global coordination rather difficult and/or degrade individual performance levels.

3. A Model for the Coordination of BS Member Tasks

In this section, we present a coordination model based on a decentralized scheme. We view coordination as a way to ensure coherence of group task executions in order to improve overall performance in spite of the agents limited knowledge of each other.

Coordination and execution activities are conducted by a group of agents. An agent is either task coordinator (also called coordinator) or task executor (executor). To each coordinator is assigned a task executor. A coordinator is responsible for determining the next tasks, which we call intentions, to be implemented by the associated task executor. A task becomes an intention when its future implementation is agreed upon by group coordinators. Upon completion of a task, executing agents receive from their associated coordinator the next intention to be implemented. In our model, an intention can be revoked if necessary; however a task being executed cannot be interrupted.

Coordination is conducted in an incremental fashion. Each coordinating agent determines a candidate task as its next intention. Coordinator selected tasks are submitted to other coordinators for acceptance. Whenever an agreement is reached, the process starts again for the determination of the next intentions.

The coordination model makes the following assumptions:

- A fixed number of agents is present in the model, hence no reduction or addition of coordinators or task executors is considered;
- Individual responsibilities are assigned and remain static during the coordination/execution process. No role merging is allowed;
- A relative ranking (hierarchical) of the agents is established;

- Procedures, checklists and the underlying tasks are predetermined;
- Each coordinator has some knowledge for identifying tasks relationships.

A task is characterized by the following parameters: name, type, associated agent, expected start time, expected duration, needed resources (type and amount).

The coordination process consists of two levels: The local intention selection and the distributed coordination protocol. The local intention selection is an individual process which determines the next steps. The distributed coordination protocol corresponds to the management of the interaction between coordinators in order to reach agreements.

3.1 Local Intention Selection

The local intention selection process helps answer the question: given the current status of the other agents executed tasks and intentions, what should be a coordinator's next intention? This process consists of: candidate tasks identification, relationships detection and candidate task selection.

Candidate task identification

At the candidate task identification level, the following information is available to a coordinator:

- its context-dependent tasks, formulated as plans with temporal dependencies, and its persistent tasks;
- its task currently being executed;
- its intentions already accepted by other coordinators;
- other agents tasks being executed and intentions.

The candidate tasks set is the combination of the persistent tasks, and the context-dependent tasks not violating any temporal precedence with current intentions and execution. A hypothetical start time, corresponding to the expected completion time of the last agreed upon intention, is assigned to each candidate task.

Relationship detection

The relationship detection consists of identifying the relationships between candidate tasks and other agents task executions and intentions.

Relationships between tasks [10], can either negatively or positively affect concurrent execution. Negative relationships may generate potential inconsistencies in group activities, while positive relationships may constitute a basis for improving the synergy between two individuals.

Table I: Types of task relationships used for modeling

TYPE	NAME	EXAMPLE
Negative	temporal precedence	recall(agt1) & receive-briefing-from(agt1)
	task request	Direct(ag1,determine RWstatus) & determine(RWstatus)
	common resources	consult(charts) & prepare(breifing)
	simultaneous execution	discuss-with (agt1,Wxstatus) & discuss-with (agt2,Wxstatus).
positive	favor exchange	call(agt1, status)

In our current effort to model BS task relationships, four different types of negative relationships are retained: temporal precedence, task request, common resources, and simultaneous execution. Inconsistencies arise when temporal precedence is ignored, task requests arise under hierarchical authority (inter-agent commitments), common resources are scheduling constraints to prevent overutilization, simultaneous execution necessitates interactions between officers in order to jointly assess/deliberate/agree.

The only positive relationship considered in this work is favor exchange [10] which is possible when more than one agent have the capability to execute a given task. Agreement between officers should then be sought in order to avoid redundancies. In general, it is expected that favors are made by inferiors to accomodate superiors (task delegation); however in this model, task favors are assigned following dynamic group deliberations.

The identification of relationships is based on the task type, the expected start time and duration of task, and on the nature and amount of resources needed for execution.

Candidate Task Selection

Candidate task selection is a computation done locally by each coordinator in order to determine the next intention to be proposed to other coordinators. This selection is based on the list of candidate tasks previously identified and their detected relationships. The intention choice process consists of a conflict avoidance step and a task ranking step.

Conflict avoidance consists of either candidate task elimination, or candidate task parameter modification depending on the relationships. Temporal precedence and simultaneous execution conflicts can be avoided by shifting candidate task start time. If avoidance is revealed impossible due to prior selections, a group intention revision procedure must be initiated as explained in section 3.2. Common resources constraints can be relaxed by either reducing the amount of resources or shifting start time. Favor tasks are kept as candidate tasks to be ranked; however in the case where they are being executed by another agent, they are eliminated since task executions cannot be revoked. Candidate tasks involved in a request relationship are kept for later ranking.

Candidate intention selection is represented as a decision problem based on a multicriteria ranking of the modified candidate tasks, resulting from the conflict avoidance step. In this current model, task ranking is done according to the following criteria:

- the nature of the tasks;
- the type of identified task relationships;
- a cost assigned to the task modifications;
- the agents involved in the relationships.

A local intention selection based on such criteria can be seen as a compromise between what is locally most preferable and what is the least constraining for group activities. One difficulty associated to this formulation is the determination of trade-offs among various hypothetical situations. For example, one would have to determine if it is more preferable to do a favor to a superior or anticipate a request relationship with inferiors. Similar considerations should be given to ensure that execution of the persistent tasks is eventually favored over context dependent task. Such trade-offs can only be determined in a domain specific study and require for further investigations.

3.2 Distributed Coordination Protocol

When a candidate intention is locally selected, it is submitted for other coordinators approval, who upon receipt can either:

- accept sender intention;
- accept sender intention and change their own intentions;
- accept and relax their own intentions to accomodate the sender, through a temporal shift or a resource reduction modification.
- refuse sender intention (bring sender to change its intention);

If a proposal is refused, the proposing agent initiates a new local selection process using alternative candidate intentions. Counterproposals by the receiving coordinator are not admissible in this model, since they can only be made when extensive knowledge of other agents is available.

An agreement is reached when a group of intentions is accepted by all agents. However, general acceptance of additionnal intentions or modifications may require a global revision of previoulsy accepted ones, since they may create new conflicts. Such discrepancies may result from the asynchronous nature of inter-agent communications. It is also possible that execution time delays postpone the detection of negative and positive relationships. In such cases, a group intention revision procedure must be declared.

A revision procedure forces the group to backtrack (backjump) over their already determined intentions and reinstantiate new ones where necessary. This backtracking (backjumping) can be seen as a search over the possible group intention space. A group revision can be declared by any coordinating agent whenever a deadlock arises. For example, the instantiated inversion of two tasks involved in a temporal precedence creates a deadlock.

To ensure that coordination can be reached at a given time while limiting the expected duration of deliberations during group revision, it is desirable to take advantage of the relative hierarchical ranking among coordinators in order to resolve conflicts [2]. Control over intention revision is sequentially delegated to coordinators following a descending path along the group hierarchy (in our case, the BS hierarchy). The revision procedure is initiated by the highest authority which performs a local selection intention. The selected candidate intention is then broadcasted to all inferiors for acceptance based on their own intentions. When the highest authority intention is agreed upon by all inferiors, this intention is kept constant and the control is passed on to the next ranked authority which repeats the same steps with its inferiors. The process terminates when the control reaches the bottom of the hierarchy ranking.

4. Discussion

A general outline of the distributed coordination model has been presented for Battle Staff group activities. Application of the proposed scheme to realistic examples of Battle Staff operations, and comparison with other proposed distributed coordination scheme [1] [2] [9] and local intention selection are being conducted.

Coordination of group procedures is often modeled as a two-phase process: coordinate all tasks first and then submit the result for task execution. Such a scheme cannot be adopted in dynamic environments involving uncertainty. Trade-offs between complete coordination and immediate execution remain to be studied. Such trade-offs, representing the horizon of intention coordination, are largely dependent on the status of the environment and the progression of group activities. Task refinement and definition will be taken into account in further coordination studies of BS member activities.

With such a distributed scheme, can specific task combinations be imposed on a group of coordinators? Integrity of the authority among coordinators directly depends on the selected combination of parameters such as the relative importance of coordinating agents, the preference orders over tasks and relationships, as well as the frequency of intention revision.

5. Conclusion

A group coordination model of Battle Staff activities, a description of the military context and of the associated decision aid environment have been presented in this paper. The coordination model is designed to favor the selection of intentions locally. The selection process is subject to a series of relationships among group intentions. We proposed five relationships to describe task dependence. Selected intentions are then submitted for acceptance to other coordinating agents. Coordination is reached when proposed intentions are agreed upon by all other coordinators. The distributed coordination protocol also provides a group intention revision procedure to resolve conflicts over sequences of individual intentions. Group intention revision is based on the relative ranking of the coordinators, hence taking advantage of the functional hierarchy structure of organizations like the NORAD ROCC Battle Staff.

As explained in the discussion, future work will look at the validation of the proposed coordination scheme on realistic examples of Battle Staff operations and the incorporation of tasks refinement/definition to the proposed model.

Acknowledgement

The author would especially like to thank Stephen Woods and Irene Abi-Zeid for providing helpful comments relative to this work.

References

1. S. Conry, K. Kuwabara, V. Lesser, "Multistage negotiation for distributed satisfaction," IEEE Trans. SMC, vol. 21, no. 6, pp. 1462-1477, Noc/Dec 1991.
2. E. H. Durfee, T. A. Montgomery, "Coordination as distributed search in a hierarchical behavior space," IEEE Trans. SMC; vol. 21, no. 6, pp. 1363-1377, Noc/Dec 1991.
3. A. Rao, M. Georgeff, "Intentions and rational commitment," Technical Note 8, AAII, February 1990.
4. I. Greif (ed.), "Computer supported cooperative work: a book of readings," San Mateo, CA, Morgan Kaufmann, 1988.
5. M. Klein, "Supporting conflict resolution in cooperative design systems," IEEE Trans. SMC; vol. 21, no. 6, pp. 1379-1389, Noc/Dec 1991.
6. T. Malone, "Modeling coordination in organizations and markets," in Readings in Distributed Artificial Intelligence, pp. 151-158, 1988.
7. S. Minton, M. Johnston, A. Philips, P. Laird, "Minimizing conflicts: a heuristic repair method for constraint satisfaction and

scheduling problems," Artificial Intelligence; vol. 58, pp. 161-205, 1992.

8. J. Rasmussen, B. Brehmer, J. Leplat, (eds.) "Distributed Decision Making: Cognitive Models for Cooperative Work," Wiley, Chichester, England, 1991.

9. K. Sycara, S. Roth, N. Sadeh, M. Fox, "Distributed constraint heuristic search," IEEE Trans. SMC; vol. 21, no. 6, pp. 1446-1460, Noc/Dec 1991.

10. F. von Martial, "Coordinating plans of autonomous agents," Springer-Verlag, Berlin Germany, 1992.

Aurora: A Multi-Agent Prototype Modelling Crew Interpersonal Communication Network

Denis Gagné, Georges Nault, Alain Garant, Jocelyn Desbiens
Groupe de Recherche en Intelligence Artificielle Distribuée (GRIAD)
Collège militaire royal de Saint-Jean
Richelain, (Québec), Canada, J0J 1R0

{dgagne,nault,garant,desbiens}@cmr.ca

#149702

Abstract

We present the implementation details of a multi-agent prototype modelling crew interpersonal communication network during the completion of a military mission aboard the anti-submarine aircraft AURORA. Based on the ACTOR computational model, this prototype is implemented on a distributed memory parallel machine, enabling several autonomous knowledge based systems and dynamic objects, executing simultaneously, to freely interact to solve complex problems. An appreciable gain in performance, attributed to parallelism, is added to the power of knowledge based systems applied to specific aspects of military operations.

We show that complex problems such as the completion of military mission can be modelled within a multi-agent environment. We briefly discuss the numerous military applications of such modellization, in particular, a new concept in simulation training that we call Virtual Role Playing.

Introduction

More and more organizations are starting to recognize the beneficial merits and strengths of group dynamic. The expected reward being to achieve holistic results from team cohesiveness (*i.e.* results from the whole greater than the sum of the individual parts). Focus within organizations is therefore rightfully shifting toward communication relationships among individuals rather than strictly on the individuals themselves. *Interpersonal communication networks* within teams are being recognized as a central factor driving their efficiency. Based on these observations, and several others, an emerging requirement to train the "team concept" is universally being identified from many different occupations (*e.g.* civil aviation, military, *etc.*). Our research in Distributed Artificial Intelligence (DAI) has lead us to propose an innovative solution to the team concept training issue.

We have been studying systems which enable relatively independent heterogeneous agents to cooperate in solving specific problems that require their combined abilities (known as multi-agent systems). The parallel between our research and team training quickly became evident. By simple analogy, teams can be considered as being multi-agent systems (and *vice-versa*). Caution is advised in making this analogy as the general multi-agent paradigm is a concept still in its infancy with many intricate problems yet to be solved. It is nevertheless possible to define certain classes of problems for which solutions are within the realm of present technology. One such class of problems is created when applying the multi-agent paradigm to certain aspects of military teams or crews.

Multi-Agent Systems as Simulated Crews

When dealing with groups, teams or crews, there is a requirement for various functions to be distributed among members. When functions are distributed among members, so is power (e.g. control, authority). As a result, groups tend to become hierarchical, and high ranking members are able to control or influence events. Such is the case with military teams and crews.

Furthermore, the expertise involved in the operational functions of members from these military teams or crews are for most specific and specialized. Present Artificial Intelligence (AI) techniques noticeably excel in these specific circumstances.

We can then by conjecture, consider a member of a crew as a relatively independent agent involved with other agents in the resolution of a complex task, namely the completion of a military mission. Military teams or crews are further characterized by relatively strict and rigorous predefined operating protocols, with the expertise restrained to a specialized domain of application and its utilization well defined and systematic. These attributes all collaborate to the feasibility of utilizing a multi-agent approach to model crew interpersonal communication networks.

Scalability

Multi-agent systems also represent, in their own right, an effective and much needed solution to the scalability problems facing today's Artificial Intelligence concepts. Attempts to develop larger and more complex rational systems have revealed many shortcomings of traditional AI. Most approaches proposed rely on a centralized architecture which degenerate ungracefully when trying to generalize it or apply it to disparate and larger domains of expertise.

Many Artificial Intelligence concepts have thus been waiting for years for the right mixture of silicon and software. We believe that such a recipe is obtained by integrating the simplicity, elegance and expressiveness of the ACTOR computational model with the exceptional processing power of distributed and parallel architectures. The result is an approach which is totally scalable at both the hardware and software level.

The ACTOR Model

The ACTOR model [Hew 73, Agh 86] is a natural extension of the object-oriented paradigm as we know it, where each actor is an active, independent entity, communicating freely with other actors. The model is in our opinion the best currently available computational model for distributed memory parallel machines.

In the ACTOR model, an actor is defined by its "script" (*i.e.* the program code of the actor). Essentially, an actor is an independent computation entity which carries out its actions in response to the different messages it receive. Effectively, an actor consists of:

- an address, with a mailbox where received messages can be accumulated;
- a behaviour, which corresponds to the actions to be undertaken by the actor in response to the current message being processed.

Just as with the object-oriented paradigm, actors are highly independent encapsulated entities stressing locality of operations. What primarily differs the ACTOR model from the object-oriented paradigm is that an actor not only involves the static program code, but also the execution environment of this code. Each actor maintains its own execution context and heap, making it an active entity. Conversely, an object in the object-oriented paradigm, does not possess its own run time environment. Actors are thus independent processes that can execute simultaneously, making the model ideal for parallel and distributed processing.

One other strength of the model is the uniformity of its message passing protocol. Actors interact on a purely local basis by sending messages to each other [Hew 79]. Each actor knows or can easily find the address of a desired correspondent. There are no limits as to the number, size or nature of the messages an actor can send.

Furthermore, new actors can be created, others can be deleted or change their behaviour, all dynamically in a totally asynchronous and independent fashion during execution. Actors can thus agglomerate to form cooperative working groups, with the properties of being transformable and dynamically reconfigurable. These groups, consist of two or more actors who interact or communicate. Actors belonging to such agglomerations will typically share at least one common goal.

The exceptional modularity and grain size flexibility of the model allows us to envision interconnectivity at virtually any system level. Actors may vary greatly in nature and size from small simple process to substantial software such as complete knowledge based systems. (See [Des 93b], a sister article in these proceedings for more details on how knowledge based systems have been submerged in Actors.) A group of knowledge based systems submerged in such an environment are then, like any other actors, endowed with the capability of exchanging information during execution allowing them to cooperate with each other in resolving complex problems. This last capacity is a most interesting property, for it is then conceivable to easily elaborate multi-agent cooperation networks with the model.

The AURORA Prototype

The prototype presented herein is a multi-agent system executing on CLAP[✍] [Des 93a], an environment implementing the ACTOR model. An earlier version of the prototype (see [Gag 94]) was implemented using a communication ring connecting the different agents to a simulation module. Each "agent" node of the ring then consisted of a communication controller and a subjacent expert system shell. Although successful in the demonstration of the concept, this earlier version suffered from all inherent limitations of such communication topology (*e.g.* limited number of messages at any particular time, *etc.*). These problems were merely artificial as they only emanated from the implementation strategy. All these implementation problems have been overcome using the ACTOR model for the present version.

The present prototype executes on a distributed heterogeneous environment consisting of SPARC workstations and 64 nodes AVX machines[♪] [Des 93b]. An appreciable gain in power is achieved by distributing the actors over the network of processors making efficient use of the available processing power. We believe this approach to be most promising in attaining the speedup and increased processing power required of large scale Distributed Artificial Intelligence (DAI) applications.

The Actors

At the highest level of abstraction, the AURORA prototype is a multi-agent system modelling the crew's interpersonal communication network. Practically, this multi-agent system emerge from an aggregate of actors who dynamically interact. Actors involved in this system are of three different "genders"[℞]:

- agents;
- sensors;
- and simulation.

Agent actors are of a substantially larger grain size than their counter parts in the system. For these are the actors that simulate the crew members of the anti-submarine aircraft AURORA. Crew duties range from piloting the aircraft, to commanding the mission, to supervising and operating aircraft's detection instrumentation and weaponry. Presently, five of the crew members are simulated by such agents in our system: the Pilot, the Tactical Navigator (TacNav), the Airborne Electronic System Operators (AESOP1 and AESOP2) and the Acoustic System Operator (ASO). In their present version, agents individually consist of an expert system shell and a distinct knowledge base consisting of generic and operational knowledge. For this purpose the CLIPS expert system shell from NASA [Lyn 92] was ported and enhanced to execute in parallel on the distributed network. A more advanced model for agents is being elaborated to replace the expert system shell. The final model should include reactive and emergent functionality combined with voice generation and recognition modules.

***Sensor* actors** act as the agents' senses and information gathering devices over the simulated operational environment. These actors play the role of certain of the onboard instruments (*e.g.* the Magnetic Anomaly Detector (MAD) and Radar) and the pilot's visual range. For validation purposes sensor actors operate in a passive mode, that is, simulation entities advise the appropriate actor when they are within the actor's range. It is our intention to later replace these actors by actual or realistically simulated instruments.

***Simulation* actors** are all entities that are either part of, or in support of, the operational environment simulation. These include, enemies (*e.g.* submarines, ships, *etc.*), a system clock, and the Aurora aircraft. Presently, enemy subs and vessels are simulated with a simple algorithm, with dynamically modifiable depth, speed, and direction. We intend to shortly introduce more advanced stochastic models and a certain level of strategic and tactical intelligence within them.

[✍] C++ Library for Actor Programming.

[♪] AVX machines are transputer based distributed memory parallel machines from Alex Informatique.

[℞] These *genders* are introduced purely for description purposes and do not imply any implementation or model particularities.

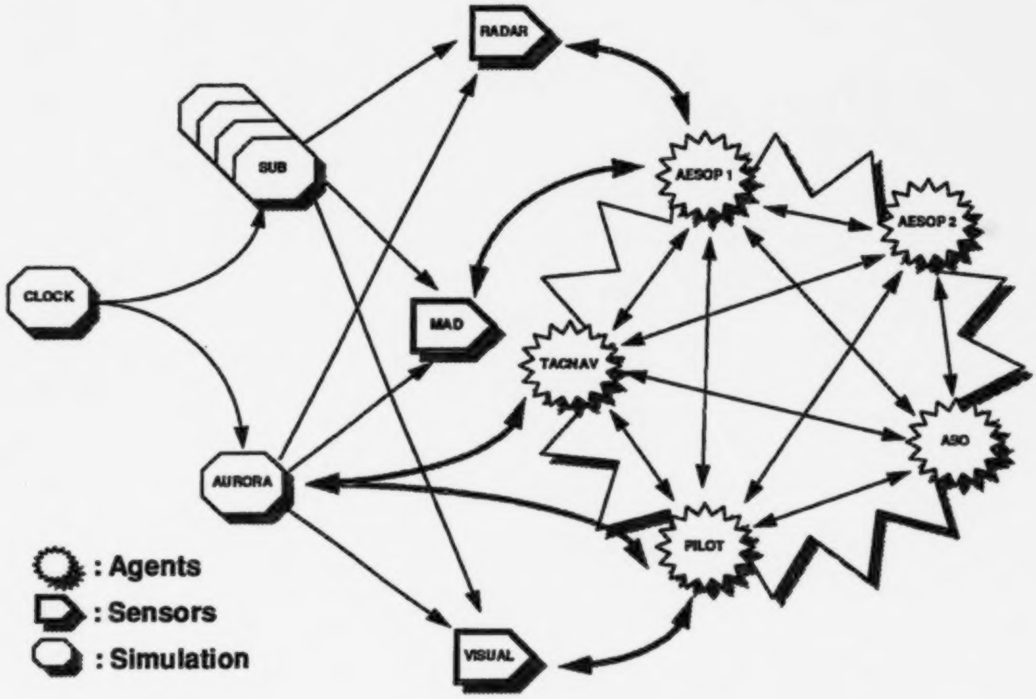


Figure 1: Communication network among actors

The Communication Network

The communication network existing between most of the actors of the systems is summarily depicted in *figure 1*. The figure only represents a snapshot, as part of the network's configuration varies dynamically. Some actors where also omitted for simplicity and clarity. It should be noted from this figure, that agents do not have any direct contact with simulation entities, except for the Aurora aircraft which they operate. Their perception of the outside world (in this case the simulated operational environment) rely solely on instrument readings and the pilot's field of view.

Actors can physically reside on a network of heterogeneous processors and communicate freely and dynamically at run time via message passing. The management of these communications is completely handled by CLAP. The interpersonal communication network of the crew is depicted by a star in the figure.

Structure of Information and Knowledge

Although the system's knowledge varies greatly and dynamically at run time, it is nonetheless classified as a *fixed knowledge configuration system*. These systems are characterised by an *a priori* distribution of tasks and functions. There is no dynamic task distribution planning in and of itself, as would be found in certain multi-agent systems. As mentioned earlier, this follows from the predefined operating protocols of military crews.

In addition, the portion of work that is classified as *intelligent* reasoning is highly localized. Communications are mostly limited to information exchanges prior to reasoning processes (or deliberative phase). Thus, deliberative phases are encapsulated in the various agents (crew members) and do not involve communication among agents.

Presently, the knowledge of the agents is mainly constituted of various procedures implementing the operational functionality of the crew members (*i.e.* operating the radar, strategic decision making, *etc.*). This expertise is represented mainly in the form of rules. There are two main classes of rules which constitute the knowledge bases of the different agents:

- internal rules,
- external rules.

Internal rules are designed to react to the an internal flow of facts. They are the rules which effectively embody the reasoning process of the individual crew members.

External rules are design to react to the arrival of new information. These rules process the observations made in the simulated operational environment and the information received from other crew members. This new information may engage internal reasoning processes.

The rules composing the knowledge bases are therefore designed in accordance with the appearance of facts. Facts have different levels of permanence in our system. Certain facts, such as the type of mission, type of vessels sought, etc., do not change during the course of a mission and each agent will therefore maintain them in its knowledge base. Other facts are non permanent or have little or no permanence, for example, event facts originating from the simulation are withdrawn from the current knowledge base once processed by the agent. Thus, agents in their current form do not retain any long-term memory of past events.

There are many challenges in systems such as this one. Moving information requires special attention. The global state of the world is distributed among the different processes being executed in the Aurora system. The truth maintenance problem among distributed agents throughout the period of time for which the system evolves therefore poses particular problems [Doy 87, Win 92]. Further, due to the nature of parallelism, the order in which messages will be received is also generally unpredictable. These aspects were not neglected in our system. At the present time, it is the responsibility of the agents to individually ensure consistency of their fact bases. Agents were also endowed with knowledge allowing them to locally prioritize certain pieces of information over others. Hence, there is absolutely no global control structure overseeing the overall Aurora system.

The Interface

The interface of the prototype is itself an actor designed to provide an external observer's view of the evolving mission (*see figure 2*). A graphical displays provides a depiction of the Aurora manoeuvring in the theatre of operation, while subjacent windows provide textual display of individual crew member's interaction with their crewmates. The interface actor acts as the intermediary between the user and the system. It permits the user to create and dynamically modify the desired theatre of operation, by positioning and configuring submarines at will in the simulated environment. The interface also allows the user to control the many parameters of the simulation (*e.g.* weather, sea state, wind direction, *etc.*) as well as the different mission objectives (e.g. ship hunting, license verification, *etc.*). The user also has the prerogative to override Aurora settings at his convenience (*e.g.* aircraft speed, direction, *etc.*).

A Typical Mission

A typical mission of the prototype will involve an initial phase of the agents (crew members) exchanging general messages containing pre-mission briefing information (*i.e.* type of mission, weather, sea state, *etc.*). The crew members then activate the mission and the Aurora can start flying its search pattern in the theatre of operation. At this time, crew members place themselves on the alert for a new signal form any of the onboard instruments (e.g. Radar, MAD, etc.). The appearance of such signal will usually trigger a cascade of reasoning, actions and communications among crew members. These operations take place until the attainment of the mission's objectives. A temporary state of balance is then again achieved and the mission can either carry on or terminate.

It is important to understand that the evolution of a mission is completely non-deterministic (*i.e.* no pre-canned scenarios or scripts), and solely based on the events taking place in the theatre of operation and the decisions and actions taken by the crew members. This is a necessary condition, if we are to present such a system as a valid experience gathering device for the teams or crews.

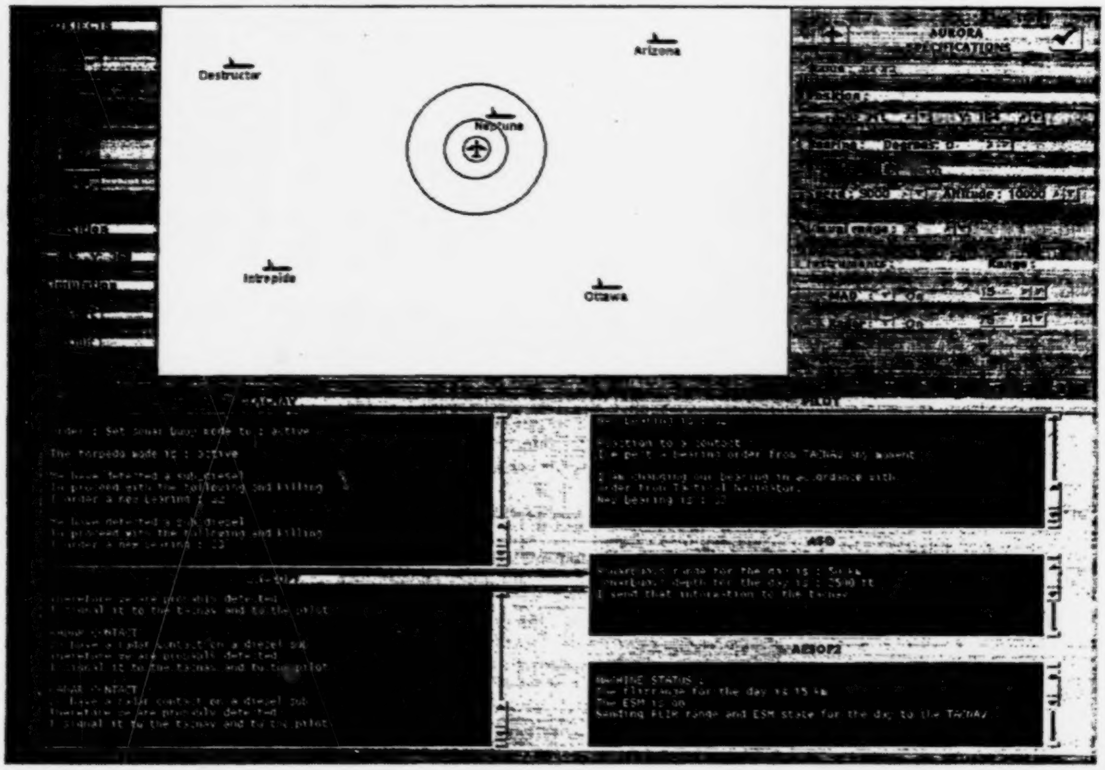


Figure 2: A screen capture of the interface during a typical mission.

Discussion

We have, to this point, presented a multi-agent system simulating crew interpersonal communication aboard the AURORA aircraft. We were very pleased by the simplicity of implementing this prototype. which we attributed, in retrospective, to the adequacy of our choice of paradigms. The ACTOR model is truly a natural lingo from which multi-agent systems can clearly and easily be described and controlled. Further to its expressiveness, the model allows us to tap on the processing power made available via distributed and parallel processing. The open architecture assures the illimitable scalability of the approach presented.

We would now like to briefly elaborate on how our approach could be utilized to train the crew concept.

The Virtual Role Playing Concept

We have proposed a new simulation training concept that we call *Virtual Role Playing* [Gag 93]. In its ultimate form, Virtual Role Playing will provide a simulated environment within which the operational functionality of each crew member will independently be simulated by an autonomous artificial agent during the completion of a mission. As a result, it will be possible to remove any number of these artificial agents and replace them by actual crew members therefore providing training to incomplete crews. Virtual Role Playing will thus provide vital crew concept training to individuals and stimulate crew cohesiveness. This cohesiveness will be achieved by allowing individuals to learn to adapt their behaviour in response to the need of the crew, via crewmates monitoring (real or virtual) during the progression of a mission.

Military Training Applications

Three generic military training areas [Thu 91] are obvious candidates for the research efforts and results presented in this article:

- Combat Mission Training;
- Mission Preview;
- Mission Rehearsal.

Combat Mission Training. The purpose of this training is to increase crew's effectiveness in a variety of simulations and not just the one specifically represented by the training scenario. In such training situation, crews are trained to reach an awareness of the dynamics and controls of the equipment with which they are interacting to extract information, what their particular contribution is to the mission and how they must interact with other crew members to accomplish it. The mission, terrain and opposing forces(s) are then generic rather than specific. (This is the prime training area targeted by *Virtual Role Playing*.)

Mission Preview. Mission preview calls for crew members to understand when and where to perform, not necessarily how to perform. In such virtual crew mode, feedback is design to elaborate and refine the plan and tactics of the mission, or even to reassign tasks based on workload evaluations. It allows "what if " hypotheses about the mission. (Note that a more advanced and enhanced version of the Aurora prototype as described in this article would fall within this category.)

Mission Rehearsal. Mission rehearsal is for conducting trial performances. It requires that the simulation represents to the highest degree possible the terrain, mission and opposing force(s) for a specific situation.

We plan continue to explore these different military training avenues while trying to elicit diverse specific applications for our work.

Conclusion

We have presented above a multi-agent prototype modelling the crew interpersonal communication network during the completion of a mission onboard the Aurora aircraft. Although individual agent's knowledge bases were very simple and purely based on non-classified information, the level of realism attained indicates that complex problems, such as the completion of a military mission, can be modelled with a multi-agent environment. Further, our research confirms that such systems are feasible within the realm of today's technology. We must nevertheless admit that our task was somewhat simplified. Military tasks being *a priori* well defined and divided, and the operational interaction protocols within teams or crews being for most strict and systematic, the complexity of the multi-agent paradigm was greatly reduced allowing rapid prototyping of such application.

The resulting system represents a demonstration of concept prototype for the concept of Virtual Role Playing. Virtual Role Playing is a new concept in simulation training that has a very broad range of applications that we have yet begun to fully explore. The Project Management Office (PMO) for the AURORA Update has just requested a concept study for the application of Virtual Role Playing for the In Flight Training Program (IFTP) of the AURORA, confirming the requirement for this type of training and the applicability of our approach. We finally concluded the article with some of the more evident classes of military applications for such modellization.

Acknowledgment

The authors would like to recognize the support from the members of the CRID at CMR. In particular, we would like to thank Pierre Raymond and Martin Lavoie for their expertise and programming skills.

References

[Agh 86] Agha, G. *"ACTORS: A Model of Concurrent Computation in Distributed Systems."*, The MIT Press, Cambridge, Massachusetts, 1986.

[Des 93a] Desbiens, J. et al., "CLAP: Un système de programmation objet pour machines parallèles à mémoire distribuée.", In *ICO Québec*, Vol. 5 , No 2, 1993.

[Des 93b] Desbiens, J., Toulouse, M. & Gagné, D., "CLAP: Une implémentation du modèle Acteur sur réseau hétérogène." In *DND Knowledge based Systems/Robotics Workshop*, Ottawa, Ontario, 1993.

[Doy 87] Doyle, J., "A Truth Maintenance System." In *Readings in Nonmonotonic Reasoning.*, Edited by M.L. Ginsberg, Morgan Kaufmann, Los Alto, California, 1987.

[Gag 93] Gagné, D., "Training the Crew Concept Via Multi-Agent Systems." GRIAD Technical Report, 1993.

[Gag 94] Gagné, D., Desbiens, J. and Nault, G., "A Multi-Agent System Simulating Crew Interaction in a Military Aircraft." To appear in *Proceedings of the 2nd World Congress on Expert Systems*, Lisbon, Portugal, 1994.

[Hew 73] Hewit, C.E., Bishop, P. and Steiger, R., "A Universal Modular ACTOR Formalism for Artificial Intelligence." In *Proceedings of the 3rd IJCAI*, Stanford, California, 1973, pages 235-245.

[Hew 79] Hewit, C.E., "Control Structure as Patterns of Passing Messages." In *Artificial Intelligence: an MIT Perspective, Vol. 2*, The MIT Press, Cambridge, Massachusetts, 1979.

[Lyn 92] Lyndon B. Johnson Space Center, Software Technology Branch, *CLIPS Version 5.1*, January 6, 1992.

[Thu 91] Thurman, R.A. and Mattoon, J.S., "Virtual Reality: Theoretical and Practical Implications." In *Proceedings of the 13th I/ITSEC*, Orlando, Florida, 1991.

[Win 92] Winston, P.H., *Artificial Intelligence.*, Addison-Wesley, Reading, Massachusetts, 1992.

Y02329

An Expert System For Condition Monitoring

C. G. Selkirk, P. R. Roberge, and G. F. Fisher
Department of Chemistry and Chemical Engineering
Royal Military College of Canada, Kingston, Ontario, K7K 5LO

K. K. Yeung
Defence Research Establishment Pacific, FMO Victoria, B.C., VOS 1BO

#149704

Abstract

For over twenty years the Canadian Forces (CF) have been interested in monitoring its operational equipment's performance and health. During this time many condition monitoring (CM) techniques have been developed to scrutinize the various aircraft systems in use. One of these methods, filter debris analysis (FDA), evolved in response to the introduction of fine (10 micron) filtration systems which limited the usefulness of conventional spectrometric oil analysis (SOA). FDA is at present limited by time requirements and the subjectivity of results. In order to counter these drawbacks, an expert system is being developed which will aid filter debris analysts in identifying wear debris and wear level trends, and provide the analyst with references in an attempt to standardize results. Once completed, this expert system will provide a foundation for designing a broader expert system which will incorporate other CM techniques currently in use. This will provide the condition monitoring analyst with a tool to correlate data and results from the various CM techniques, thus increasing the utility of each individual method of analysis. This paper will introduce FDA and then outline the development of the FDA expert system and its analytical rule base.

Introduction

Condition Monitoring

Over the past several decades the monitoring of a system's health and various wear rates has become of primary importance to maintenance managers in both industry and the Canadian Forces (CF). The savings incurred by reducing system downtime, increasing reliability, improving fault analysis, and decreasing superfluous preventive maintenance make condition monitoring (CM) appealing to civilian industry [Mathew, 1987] [Thomson, 1991] [Vigreux, 1991]. These savings combined with the more important ability to reduce the risk of damage to equipment or injury to personnel make CM the focus of many new maintenance philosophies in the CF [Dominique, 1987] [Abu-Hakima, 1989] as well as in the United States Air Force (USAF) [Schroeder, 1992] [Rock, 1993].

Early work at the Defence Research Establishment Pacific (DREP) led to the involvement of the CF with CM through the introduction of the spectrometric oil analysis program (SOAP) over twenty years ago. CM, non-destructive testing (NDT) and scheduled and unscheduled inspections combine to give the Air Force's maintenance managers an abundance of information from which they can deduce a system's serviceability. Presently there are two main CM Techniques in use by the Air Force, SOAP and vibration analysis (VA). These techniques are supplemented by other techniques such as ferrography and filter debris analysis (FDA). With the procurement of modern aircraft systems, the development of new integrated CM techniques will likely continue to ensure that the needs of maintaining advanced systems are met. An example of this ongoing development process can be obtained by considering the development of FDA.

FDA Development

In oil-wetted aircraft machinery fitted with coarse (>50 micron) filtration systems, sufficient fine particulate wear debris can pass through the filter to enable meaningful spectrometric oil analysis (SOA) of the oil sample composition for off-line condition monitoring. The use of fine (10 micron) filters in the F404 engines of the CF-18 aircraft has severely limited the usefulness of oil analysis techniques because the filters remove almost all debris of significance from the oil. DREP looked into this problem and consequently developed FDA as an alternative off-line monitoring procedure for the CF-18 F404 engines. In addition, FDA can supplement conventional oil analysis techniques for condition monitoring of coarse filtration machinery by evaluating the accumulated wear debris generated between filter changes [Yeung, 1993].

The FDA procedure begins with the removal of a system's filters (this should be scheduled at regular intervals to aid in wear trending during the analysis of the filters). Filters are then cleaned by one of a variety of methods (which depends on the system/filter type) and the filter debris is gathered on a 0.8 micron (AA Type) Millipore membrane which is subsequently allowed to dry overnight. The microscope analysis of the debris follows, during which the analyst takes note of the debris characteristics [Dominique, 1988]. Figure 1 illustrates the information requirements for the component condition assessment (e.g. light background deposit). The characteristics required for large particle identification are shown in Figure 2 (e.g. particle is grey and magnetic).

The results obtained from the FDA technique

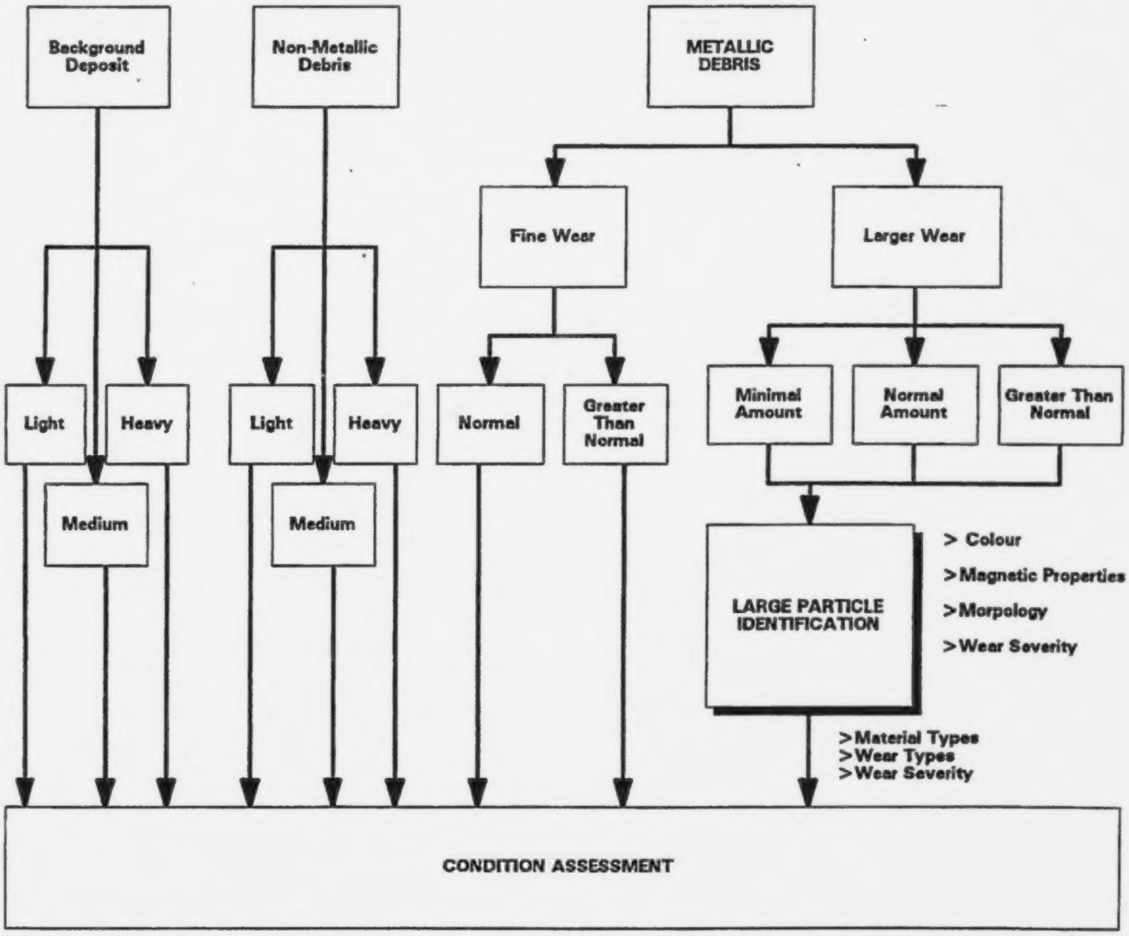

Background Deposit
Non-Metallic Debris
METALLIC DEBRIS
Fine Wear
Larger Wear
Light
Heavy
Medium
Light
Heavy
Medium
Normal
Greater Than Normal
Minimal Amount
Normal Amount
Greater Than Normal
LARGE PARTICLE IDENTIFICATION
> Colour
>Magnetic Properties
>Morpology
>Wear Severity
>Material Types
>Wear Types
>Wear Severity
CONDITION ASSESSMENT

FIGURE 1: General Debris Information For FDA

give insight into the type of metal present (e.g. Cu or Cu-based), the type of wear that is occurring in the system, and the general debris composition (e.g. mainly non-metallic fibre debris). Although it is not possible to determine the exact composition of alloys through visual inspection, this technique allows the analyst to determine when such information should be gathered due to excessive amounts of a type of debris. Presently, alloys are commonly identified by using a scanning electron microscope (SEM); however, there is work underway to study other possible techniques, such as quantitative filter debris analysis (QFDA) [Fisher, 1991].

Although FDA is still mainly used only to supplement results from oil and chip detector analysis [Dominique, 1987], it has been occasionally used in condition monitoring of the following aircraft systems:

a. CH124 Sea King - Main Gear Box and T58 Engine;
b. CF188 Hornet - F404 Engine;
c. CC115 Buffalo - Speed Decreaser Gear Box and T54 Engine;
d. CH135 Twin Huey - Main Gear Box; and
e. CH113 Labrador - T58 Engine.

As is the case with most CM techniques, using FDA as a stand alone method of analysis does not maximize the utility of the debris information gathered. When FDA results are combined with information gained from VA, for example, the overall perception of the system state is enhanced, and suspect components can be readily identified. Although it has proven its analytical usefulness, the main drawbacks of FDA are that it is time-consuming, it is manpower inefficient, and the interpretation of debris levels is subjective.

Improving Upon FDA

To rectify these problems, DREP is presently investigating the use of computer-aided image analysis to interpret the debris found on filters [Yeung, 1993] with the ultimate goal of incorporating the FDA results into an expert system for tracking the condition of oil-wetted equipment. In support of such an approach, there are currently many examples of expert systems applications in similar areas that have led to increased efficiency and reliability [Toms, 1991] [Abu-Hakima, 1989]. Some of the reasons for integrating an expert system into FDA are as follows:

a. To have the ability to store analytical experience in a Knowledge Base;
b. To attempt to reduce the subjectivity of results;
c. To shorten the time required for analysis; and
d. To correlate results from other CM techniques to aid the decision making process.

Expert systems designed to aid (not necessarily replace) analysts have already been successful in other areas, for example, in safety and reliability analysis [Grant, 1989] and in the maintenance of electronic switching stations [Kim, 1988]. These accomplishments further promote the development of an FDA expert system. The focus of this paper is to describe the project to develop the FDA expert system as well as to investigate further applications for expert systems in CM.

Expert System Development

The expert system development cycle has been broken down into the following stages:

a. Define system goals;
b. Selection of development tool;
c. Initial prototyping of system;
d. Prototype verification;
e. Updates to prototype as required with subsequent verification; and
f. System introduction.

The project has currently passed the first two stages and is well into the initial prototyping phase. As the project continues, any new system requirements which come to light will be implemented into the system. The final phase will be expanded upon as the project develops.

System Goals

As described in the introduction, there are several goals which can be achieved through expert system applications. The goals for the FDA expert system also include:

a. Correct identification of wear debris;
b. Correct identification of wear modes; and
c. Expert guidance in determining problem areas and solutions.

These combined goals, when met, will improve the quality of the information gained through FDA, as well as making this method of analysis more accommodating to the support requirements of other methods of CM.

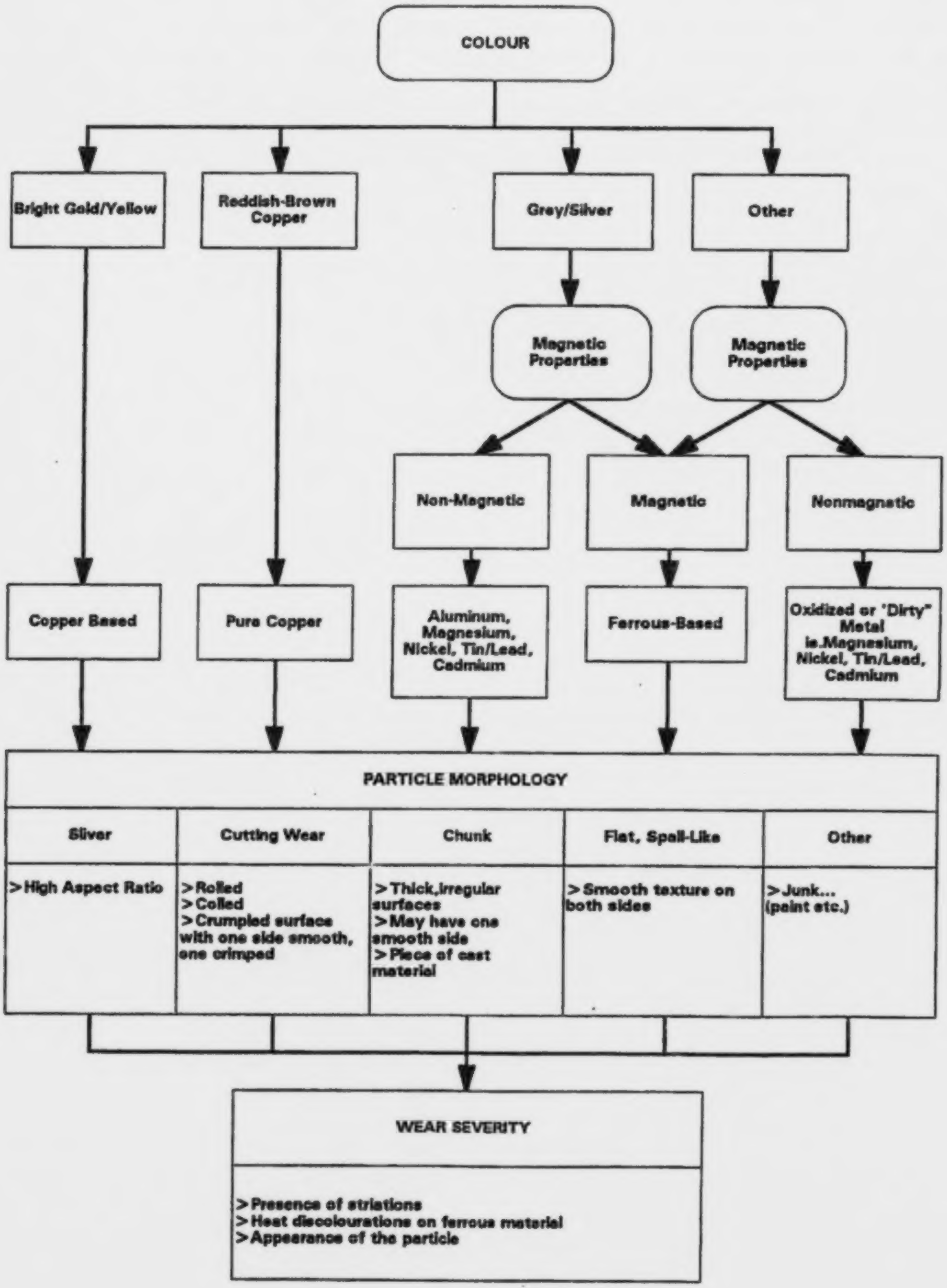

COLOUR
Bright Gold/Yellow
Reddish-Brown Copper
Grey/Silver
Other
Magnetic Properties
Magnetic Properties
Non-Magnetic
Magnetic
Nonmagnetic
Copper Based
Pure Copper
Aluminum, Magnesium, Nickel, Tin/Lead, Cadmium
Ferrous-Based
Oxidized or "Dirty" Metal ie.Magnesium, Nickel, Tin/Lead, Cadmium
PARTICLE MORPHOLOGY
Sliver
Cutting Wear
Chunk
Flat, Spall-Like
Other
>High Aspect Ratio
>Rolled
>Coiled
>Crumpled surface with one side smooth, one crimped
>Thick,irregular surfaces
>May have one smooth side
>Piece of cast material
>Smooth texture on both sides
>Junk... (paint etc.)
WEAR SEVERITY
>Presence of striations
>Heat discolourations on ferrous material
>Appearance of the particle

FIGURE 2: Large Particle Identification Characteristics For FDA

Programming Tool

The choice of a programming tool for use in developing the system was based upon several criteria. The first factor to consider was the "user friendliness" of the proposed expert shell. The expert system shell must be readily comprehensible as well as being capable of guiding the user/analyst through the inspection process. These are requirements that go hand in hand with the need to make the system thorough and dynamic. In this area, Visual Basic™ (VB) offered the ability to easily design front-ends using graphical user interfaces (GUIs), while maintaining extremely versatile application capabilities. GUIs are created by drawing objects which can then be assigned properties to define their appearance and behaviour. Code is then written which will respond to events that affect the objects in the interface. For example, clicking on an "execute" icon will run code for a desired function, while clicking an "exit" icon will save data and leave the system.

The next consideration was the trade-off between programming complexity and consequent benefits. The language to be used should not be excessively complicated without just cause, for reasons of ease of development and feasibility of future changes/upgrades. VB has proven to be an easy language to learn with numerous capabilities that aid in the prototyping process. As well, by programming in a moderately high level language like VB (versus lower level assembly-style C), the future editing will be facilitated by the easy-to-follow logic of event driven code.

Another factor which influenced the decision was that of compatibility. VB is a Windows™ application which has the capability to converse with other Windows applications through the use of multiple-document interfaces (MDIs), object linking and embedding (OLE), dynamic data exchange (DDE), dynamic-link libraries (DLLs), and more [Microsoft, 1992]. This gives VB a multi-role ability that is very appealing for graphical and database requirements. For these reasons, VB has been chosen as the programming tool to develop the expert system shell and knowledge base, although there are several other similar tools available which were found to be reasonably suited to the task (e.g. C++ for Windows).

Expert System Design

The FDA expert system is being designed to mimic the logical reasoning of an experienced analyst in determining the type of metallic material and the wear condition of the machinery based on the colour, morphological attributes (Figure 2), and surface texture of the wear particles. By automating this process, and incorporating the expert analyst knowledge base, the system will help the analyst/user make quicker, well informed and less subjective decisions in FDA.

One of the ongoing processes included in the initial prototyping of the system is that of knowledge acquisition. The three principal approaches which can be used are deductive (fault tree analysis), inductive (failure-mode-effect analysis) and experience-based (analysis of data regarding past failures) [Schmidt, 1989]. The FDA expert system's approach is that of an experienced-based system which will reference current wear trends with past experience to detect problem areas. Some possible methods for incorporating this knowledge and experience into the system include using rough sets [Szladow, 1993], decision trees [Pepper, 1993], or a rule base [Schmidt, 1989]. The FDA expert system uses a rule-based technique in which the rules come from querying the expert analyst about procedures, wear trends, wear patterns and resultant problem areas. Although it is impossible to be exhaustive in knowledge acquisition, the techniques which have been chosen will be able to assimilate much of an analyst's experience into the knowledge base. Improvements to the knowledge base will be achieved by obtaining opinions, information and guidance from as many experienced analysts as possible.

Due to the present subjectivity of results in FDA, a photographic database will also be included in the expert system in order to give the analyst a reference to compare to when analyzing debris. The database will include examples of the various wear rates (e.g. normal or excessive) as well as colour and morphological classification examples (e.g. sliver or cutting wear). These examples will be further classified by system/component type and parent aircraft type since each aircraft system will have different wear rates and debris associated with them. By requesting necessary debris information, allowing concurrent access to the photographic database (for comparison) and by including extensive Help functions throughout the process, the FDA expert system is expected to meet its design goals.

Future Applications

Completion of the FDA expert system will provide a foundation upon which expert systems for other CM techniques may be built. The incorporation of other CM techniques currently in use would result in an overall expert system that could imitate the reasoning of the maintenance manager as well as that of analysts for each CM technique. This hierarchical approach (Figure 3) would allow the user to perform trouble-shooting operations on systems using information from appropriate CM techniques. The time saved by being instantly advised on probable courses of action (e.g. using alternative CM technique or increasing sampling rates) could decrease the downtime of a suspect system, thereby increasing its availability.

Conclusion

As the CF considers better ways to increase reliability, maintainability and availability of its aircraft systems now and in the future, expert systems may prove to be the key. The FDA expert system currently being developed will provide a working example of how expert systems applications can aid condition monitoring analysts in determining both system state and consequent corrective actions. Judging by the progress of similar less comprehensive systems which are already in place in the Canadian Air Force [Abu-Hakima, 1989] and other Air Forces of the world [Schroeder, 1992], expert system applications will continue to be an important part of the maintenance management of aircraft systems.

References

Abu-Hakima, S., et.al. [1989] "Jet Engine Technical Advisor (JETA)", Proceedings: The 2nd International Conference on Industrial and Engineering Applications of Artificial Intelligence, pp. 154-160, June 1989.

Dominique, H. P., and Mckenzie, A. J. [1987] "Filter Debris Analysis For Sea King Helicopter Main Gear Boxes", Defence Research Establishment Pacific Technical Memorandum 87-3, March 1987.

Dominique, H. P., and Mckenzie, A. J. [1988] "A Filter Debris Manual For Sea King Helicopter Main Gear Boxes", Defence Research Establishment Pacific Technical Memorandum 88-11, July 1988.

Fisher, G. F. [1991] "Quantitative Filter Debris Analysis (QFDA) as a Means of Monitoring Wear in Oil-Lubricated Systems", RMC Department of Chemistry and Chemical Engineering Thesis, July 1991.

Grant, M., and Harvey, D. W. [1989] "A Progress Report on the Development of FIABEX", Proceedings: Reliability '89, vol. 2, pp.4c/2/1-8, 1989.

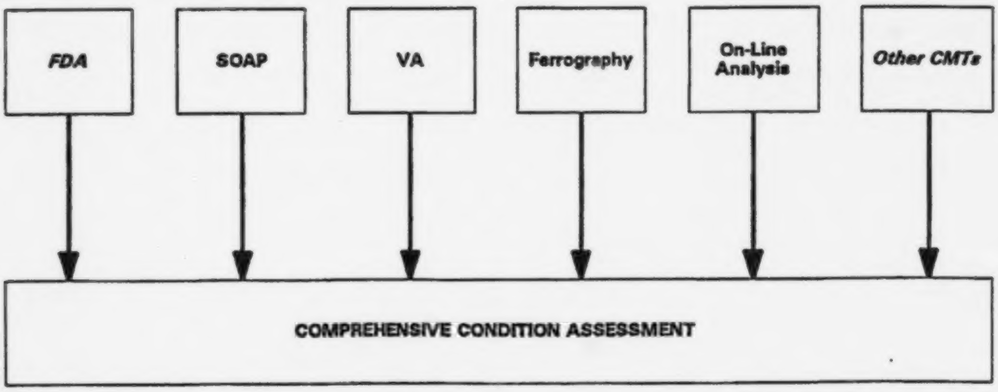


FIGURE 3: Combine FDA With Other CMTs For Complete CM Expert Sytem

Kim, J., et.al. [1988] "A Knowledge Based Message Interpretation for the Maintenance of an Electronic Switching System", Proceedings: The 1st International Conference on Industrial and Engineering Applications of Artificial Intelligence, pp. 334-40, 1988.

Mathew, J. [1987] "Machine Condition Monitoring Using Vibration Analysis", Acoustics Australia, vol. 15, pp. 7-13, April 1987.

Microsoft® Visual Basic™ Language Reference, Microsoft Corporation, California, 1992.

Pepper, J. [1993] "A Methodology for Building Knowledge-Based Systems", PC AI, vol. 7, no. 1, pp. 18-21, January/February 1993.

Rock, D., Malkoff, D., and Stewart, R. [1993] "AI and Aircraft Health Monitoring", AI Expert, vol. 8,no. 2, pp. 28-35, February 1993.

Schmidt, T. H. [1989] "Model-Based Fault Diagnosis: Knowledge Acquisition and System Design", Proceedings: The 2nd International Conference on Industrial and Engineering Applications of Artificial Intelligence, pp. 21-25, June 1989.

Schroeder, J. B., et.al. [1992] "Field Test of an Advanced Maintenance System", Proceedings: 1992 Annual Reliability and Maintainability Symposium, pp. 216-222, January 1992.

Szladow, A., and Ziarko, W. [1993], "Rough Sets: Working With Imperfect Data", AI Expert, vol 8, no. 7, pp. 36-41, July 1993.

Thomson, W. T. [1991] "Research and Development of On-line Diagnostic Monitoring Systems for Electrical Machines", IEE Colloquium on Instrumentation of Rotating Electrical Machines, pp. 5/1-17, February 1991.

Toms, D. C., and Tenner, D. [1991] "LV2Debug: A Rule-Based Tool for the Troubleshooting Process", Conference Record: Autotestcon '91., pp. 249-252, September 1992.

Vigreux, J. [1991] "Application of Condition Monitoring Techniques in Gas Insulated Substations", Electra, no. 134, pp. 46-60, February 1991.

Yeung, K. K., Luoma, G. A., and Mckenzie, A. J. [1993] "A Colour Image Processing Algorithm to Identify Copper-Based Particles In Filter Debris Samples", Defence Research Establishment Pacific Technical Memorandum 93-19, March 1993.

90C206

EXCLASS—Knowledge Based Job Classification and Evaluation

Gail E. Harris,[1] M.Sc.
RES Policy Research Inc,
Suite 501, 100 Sparks Street, Ottawa

Martin Laplante,[1] Ph.D.
RES Policy Research Inc,
Suite 501, 100 Sparks Street, Ottawa

#149705

Abstract

EXCLASS is a computer system for writing, classifying and evaluating job descriptions within the Universal Classification Standard developed for PS2000. The user interacts with the system to specify a conceptual form of a job description from which the system generates natural language text. The system consists of three modules: User Interface, Job Description and Job Evaluation. The focus of this paper will be on the Job Evaluation Module (JEM) which evaluates the sixteen elements *comprising the substantiating data. The substantiating data describes the intellectual and physical effort required to do the job, the environment affecting the work, the responsibilities of the job, and the skills and knowledge qualifications required. A set of expert system modules classify and evaluate the conceptual form of the substantiating data for the job according to the criteria defined in the Universal Job Evaluation Plan. The modules use a combination of rule based reasoning, frame based reasoning, fuzzy logic and procedural algorithms to infer a degree rating for each element evaluated. This paper describes the architecture of the system and the knowledge representation issues of the expert system.*

Introduction

EXCLASS is a computer system for writing, classifying and evaluating job descriptions within the *Universal Classification Standard* (UCS) developed for PS2000. The user interacts with the system to specify a conceptual form of a job description from which the system generates natural language text. A set of expert system modules evaluate the conceptual form according to the criteria defined in the *Universal Job Evaluation Plan* (UJEP).

The four main objectives for this system are:

1. to provide a tool to managers to support new responsibilities under PS2000;
2. to merge and automate the job description and evaluation processes;
3. to facilitate the production of comprehensive job descriptions; and
4. to ensure consistency and relativity in the evaluation of positions across the Federal Public Service.

A job description is made up of three parts: client service results, key activities and substantiating data. The *client results* specify the client and the nature of the service being provided. The *key activities*, usually five to ten sentences, describe the tasks to be performed to achieve the results but not how they are to be performed. The *substantiating data* describe the *factors* that affect the performance of the key activities. This includes details about the mental and physical effort required, the circumstances and environment in which the work is done, and the skills and knowledge that will be used. The factors are subdivided into *elements* for more focused descriptions. Each factor and element is described in the UCS and a list is provided in Appendix A. The major task of job evaluation is to find a *degree* for each element. These degrees are combined to give the *level*. Substantiating data are the major input to this evaluation.

The rest of this paper describes the software architecture, knowledge representation structures, examples of knowledge representation and inferencing for two elements, and directions for further research.

---

[1] formerly of Rockwood Informatics

System Architecture

The system is separated into three modules: User Interface Module (UIM), Job Description Module (JDM) and Job Evaluation Module (JEM).* This paper describes the JEM only. The JEM was developed in Microsoft Windows^{TM} using Kappa-PC Version 2.0.7 which combines object oriented programming and rule based programming.

The JEM contains rules and frames used in inferencing to determine the degree of each element. Both backward and forward chaining are used with the same rules. Frames are used in two possible ways (see Subsection **Frames**):

1. To organize the job description lexicon.

   A hierarchy of frames is used to represent equivalent examples of an abstract concept. During inferencing the rules search for a match between the value given in a statement instance and the specific examples of a frame, and the abstraction, that is, the parent's value, is then used in the rule clauses; or

2. To aggregate many concepts for comparisons.

   The statement instances are used to describe a frame for the current job which is then compared to some prototype frames to see how similar they are, using a distance measure.

This module also contains classes and methods for the programming required to perform evaluations and to store evaluation results. There is a class definition for each factor and each element, and class definitions for all the different statement types used by the elements. Instances of these classes are dynamically created and destroyed as the job descriptions are read from the database. The class definition for an element class includes a method for evaluating that element. The method may invoke the inference engine with different rule bases or implement the procedural logic required to evaluate the element.

The evaluation of any element is independent of all other elements. This is achieved by repeating information from one element in statements of other elements that require the same input. For example, the "nature of the service being provided" is required to evaluate the Influence element, Interaction element and others.

Knowledge Representation

The JEM uses a combination of rule based reasoning, frame based reasoning and fuzzy logic to evaluate each element. The conceptual form of the substantiating data consists of statements, which are n-tuples of attribute-value pairs. The values specified in a statement instance are used in comparisons within rule clauses or to specify a slot value in a frame.

Overall Statements

An *overall* statement describes the job as a whole. The attributes are abstractions that apply to all aspects of the activities and environment. For example, the Thinking Challenge element uses the concept of "Situations" to describe whether the challenges posed by performing the key activities are routine or non-routine. The abstraction "Situations" describes the overall job; it is an attribute of the job whose value is the same even on different occasions. A particular situation may be routine or non-routine but an overall assessment of the situations faced remains the same.

An overall statement is used when it is impossible or intractable to ask the user to describe *all* the different instances. In the above example it would be unreasonable for the job description to include details of each situation with which the job must deal.

Instance Statements

An *instance* statement describes a particular example of some aspect of the job, where that aspect may vary from time to time or for each occurence. The attributes are abstractions that describe information about the activity or concept that are required to evaluate it. For example, the Communications element requires information about the different kinds of communication that occur in the job. The attributes are used to describe who the audience is, the knowledge level of the audience and the content of the information communicated.

Instance statements are combined during inferencing to obtain an overall rating of that aspect of the substantiating data. The statements are evaluated individually and the overall rating is either the maximum rating or a weighted average of the ratings. The weighting scheme is determined with the domain expert during knowledge elicitation. In some cases the weights are based on a ranking of one of the attributes of the statement, but in many cases equal weights are used.

---

*The three modules communicate via semaphores, message-passing and database records.

Frames

Frames are used both as a method of organizing knowledge in a hierarchy of abstractions and as a method to compare several aspects of a job with a minimum of rules. In the first approach, a hierarchy of frames exist as reference information created with the domain expert during knowledge elicitation. These frames usually represent groupings of possible attribute values that should be treated as equivalent concepts during rule inferencing. When a rule fires for a particular statement, the rule may include a clause which searches for the appropriate reference frame and then uses the concept represented by the frame in subsequent clauses. For example, for the Communications element there is a frame which groups the values mediating, negotiating and persuading as the concept *reach agreement*, and the rules for Communications use the term *reach agreement* rather than the specific values.

In the second approach, a system of frame classes and frame instances exists that represent examples of reference information with certain default slot values pre-determined by the domain expert. The statements for an element specify a particular frame class and a unique instance. If some default slot values inherited from the class are inappropriate, the job description will include statements specifying new slot values that better describe this aspect of the job being evaluated. During evaluation, the job's frame instance is then compared to the existing prototype frames of the frame class using a *distance* measure. The distance value is used by rules to decide if the job's frame instance is similar to a particular prototype frame, in which case the rule concludes a degree rating or the value of an intermediate attribute. Equation 1 is the formula used to calculate the distance, where m_i is the number of possible values, n_i is the index of the example frame instance's value, and x_i is the index of the job instance's value. An example calculation is given in Section **Environment Element**.

$$\frac{\sqrt{\sum(x_i - n_i)^2}}{\sqrt{\sum(m_i)^2}} \qquad (1)$$

This approach reduces the number of rules required to evaluate a job since it is now unnecessary to have a rule for each possible combination of values for the frames' slots. It is assumed (and tested) that the possible values for a slot are ranked since the difference between the indexes of the job's instance and the indexes of the prototype frame are used in the formula, that is, the larger the difference the greater contribution to the overall distance. It should be noted that the distance value does not indicate a direction. Furthermore, the thresholds used by the rules to decide if the distance value was small enough to call the frames similar, were calibrated with the domain experts.

The distance measure can be interpreted as the complement of a membership function, where the actual distance specifies the degree of membership of the job in the frame. The membership value can now be used within fuzzy rules.

Rules

The rules used for evaluation are the typical "if—then" format. If an antecedent clause uses a particular object-slot pair then that triggers backward chaining for all rules that assign a value to that variable in their conclusion.* Similarly, forward chaining proceeds until there are no more rules that use a variable in their antecedent that was previously concluded, or the goal is proven. This means that a rule can fire in many different ways if the conclusion uses many variables.

The depth of inferencing was usually three, where the depth is measured based on rule groups. That is, suppose the rules are grouped according to the variables in their conclusion, and each rule only had one variable in the conclusion that could cause it to fire (backward chaining) or that could cause another rule to fire (forward chaining). If during inferencing, either forward or backward chaining, group A rules caused group B rules to fire and group B rules caused group C rules to fire then the depth would be two.

Environment Element

The details of how the Environment element is evaluated is included to illustrate the use of frames in an actual case. The three input attributes, *frame*, *location name* and *exposure*, form one 'instance' type of statement, which causes the system to create an instance of *frame* to describe the location and it inherits the slot values of its parent class.

---

*The backward chaining algorithm does not employ unification.

The three input attributes, *location name*, *slot* and *value*, form another 'instance' type of statement which is used to modify the slot default values of the location frame created in the first statement. For each possible location frame describing the job's environment there may be many modifying statements that specify new slot values, thus overiding the inherited values. For example, suppose there is a warehouse frame with the slots and possible values for slots as shown in Table I.

Suppose also, that the default slot values for an example warehouse and for the specific warehouse described in the job are as shown in Table II. The location statement would then be the triplet *Warehouse, Booth Street, extended* for the Frame, Location Name and Exposure attributes. The two modify statements would be the triplets *Booth Street, Dust or Dirt, none* and *Booth Street, Chemicals, indirect contact* for the Location Name, Slot and Value attributes.

Table I: Slots and possible values for a warehouse frame.

Attribute	Possible Values (ranked)		
Dust or Dirt	none	some	large amounts
Fumes or Smoke	none	some	large amounts
Temperature Changes	none	some	severe
Weather Conditions	full protection	partial protection	no protection
Chemicals	no contact	indirect contact	direct contact

Table II: Slot values for the warehouse frame and for the warehouse described in the job.

Slot	Prototype Frame	Job Instance
Dust or Dirt	some	none
Fumes or Smoke	none	none
Temperature Changes	none	none
Weather Conditions	full protection	full protection
Chemicals	no contact	indirect contact

The evaluation for this scenario would proceed by comparing the Booth warehouse to the prototype frame warehouse using the distance formula stated in Section **Frames**. The result being $\frac{\sqrt{2}}{\sqrt{45}} = .21$. The rules would then determine that the frames are similar, thus concluding that the Booth Street warehouse has the same value for the *disagreeable* attribute (an intermediate conclusion) as the prototype frame. The evaluation method would then calculate an average of the exposures and a weighted average of the *disagreeable* attribute for all the locations described in the job, using the exposure as the weighting factor. The two results are used to look up the degree in a decision table.

Thinking Challenge Element

The details of how the Thinking Challenge element is evaluated are described here to further illustrate the methods and techniques described above. The input concepts and their possible values are shown in Table III. These are grouped into two statement types, where the first type is a 1-tuple whose instances contain values for the attribute *thinking process*, and the second type, an overall statement, is a 4-tuple whose instance contains values for the rest of the attributes listed.

The evaluation proceeds in three steps: first a rating is established for each thinking process by finding the frame in which the particular value was defined. The evaluation method averages the results to find an overall rating for the thinking process. Second, the inference engine uses backward chaining to determine the predictability attribute and the difficulty of challenges attribute (see Figure 1), and finally the degree rating is determined by looking up the rating in a decision table of difficulty vs. predictability.

Conclusion

The task of developing the knowledge representation and inferencing for this project presented some unique challenges and resulted in the integration of several AI techniques. The knowledge elicitation process indicated the need for two types of declarative statements, instance statements and overall statements, which co-exist within an element. The knowledge inferencing proceeds differently for the two statement types, but eventually the job is described by overall statements only, from which a final degree is deduced. The meta-knowledge, the rules and procedures used to transform the instance statements into overall statements, is structured and therefore the possibility of defining a language for the meta-knowledge is a subject for further research. The use of frames and fuzzy logic concepts is a novel integration of these technologies and is also a subject for further research.

Table III: Input concepts for Thinking Challenge element.

Thinking Processes	Nature of Work	Scope	Situations	Reasoning Logic
instances	overall	overall	overall	overall
create	managerial	local	routine	well understood
develop	technical	regional	non-routine	partially unknown
modify	procurement	national		unknown
evaluate	functional manager	international		non-existent
analyse	labour			
comprehend	...			
...				

EXCLASS advises managers on job analysis, helping in the writing of concise, complete and unambiguous job descriptions. It consistently evaluates these job descriptions, providing explanations and rationales for each rating or the reason why a job could not be evaluated. EXCLASS successfully integrates natural language generation with knowledge based programming, including rule based reasoning, frame based reasoning and fuzzy logic.

Hardware and Software

All that is required to use the full EXCLASS system is a personal computer and Microsoft Windows^{TM}. All job information is stored in SQL databases. The entire system fits in about 3 MB. Four different software packages were used for development: Kappa-PC Version 2.0.7, LPA Prolog for Windows Version 1.42, QELib and DBase III+, and a special purpose DLL written in Turbo Pascal.

Research Directions

The knowledge contained in the EXCLASS prototype allows it to evaluate jobs in the purchasing and materiel management domains. Further research is planned to expand the domain of jobs that can be evaluated. The existing knowledge incorporates some concepts from fuzzy set theory which will be further developed as new knowledge and new domains are added. Fuzzy logic could be useful for quantifying certain attributes (eg. exposure), combining the results of instance statements, and comparing frames.

The knowledge acquisition for EXCLASS is slow and labour intensive, as is the case with most expert systems, however, the knowledge is well structured. Further research will be pursued in defining a language that expresses the knowledge and meta-knowledge required to evaluate an element. A compiler and a knowledge acquisition tool for this language is also part of the continuing research for this project.

References

[1] Rockwood Informatics Corp., "EXCLASS Job Evaluation Prototype: Physical Design Document", 1992.

[2] RES Inc., "EXCLASS Job Evaluation Module Implementation Plan", 1993.

[3] Treasury Board Secretariat, *Universal Classification Standard*, 1993.

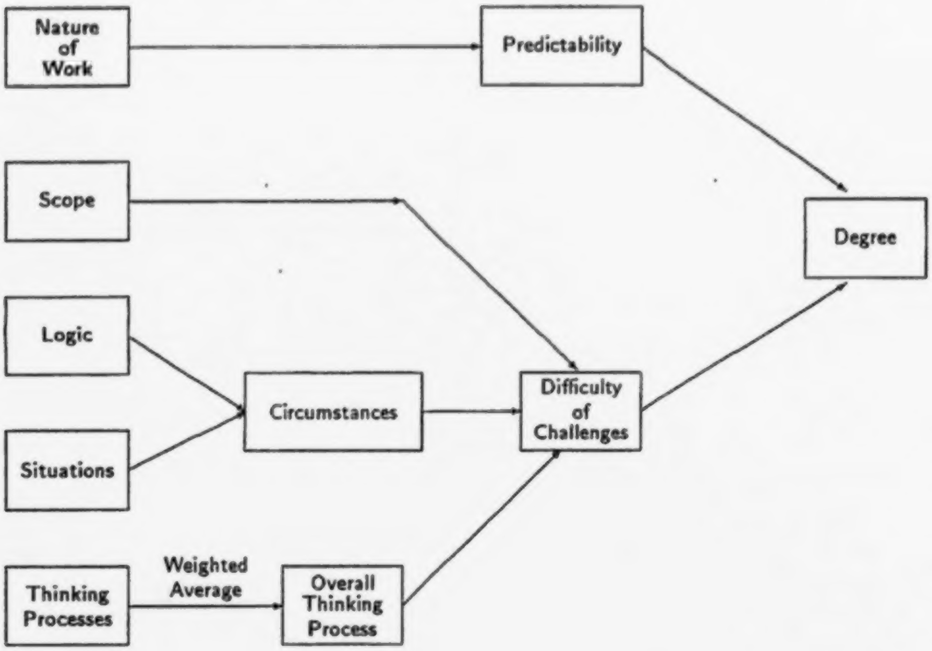

Nature of Work
Predictability
Scope
Degree
Logic
Circumstances
Difficulty of Challenges
Situations
Thinking Processes
Weighted Average
Overall Thinking Process

Figure 1: Inference chain for thinking challenge.

Appendix A: Factors and Elements

Service Delivery

1. Interaction
2. Influence
3. Thinking Challenge
4. Physical Demands

Care and Responsibility

1. Responsibility for the Work of Others
2. Responsibility for the Care of Individuals
3. Responsibility for Financial Resources
4. Responsibility for Technical Resources

Working Conditions

1. Environment
2. Risk to Health

Skills and Knowledge

1. Context
2. Acts and Regulations
3. Theories and Principles
4. Methods, Techniques and Practices
5. Communications
6. Physical Dexterity

90C 211

Concept-based composition of job descriptions with automatic text generation

Tatiana Korelsky & David E. Caldwell
CoGenTex, Inc.
5911 Dolbeau, Montreal, PQ H3S 2G1
E-mail: tanya@cogentex.qc.ca, ted@cogentex.qc.ca

#149708

Abstract

The EXCLASS System, being developed for the Treasury Board Secretariat (Under the auspices of the Artificial Intelligence Research and Development Fund), automates the description and classification of jobs in the public service. The Job Description Module (JDM) of EXCLASS supports the analysis and creation of job descriptions, and the Job Evaluation Module (JEM) performs automatic evaluations of them.

One major innovation of the EXCLASS project is to support the creation of job descriptions from conceptual elements, rather than ordinary text. Once conceptual forms are created, the corresponding textual forms are generated automatically. Text generation is closely linked with the conceptual editing process, so that users have constant feedback of textual forms to confirm the accuracy and coherence of their conceptual choices.

An advantage of describing jobs on a conceptual level is that it facilitates automatic processing of the content of job descriptions—most importantly, jobs can be evaluated by the JEM by comparing them on a conceptual level with established standards. Conceptual representations also permit users of the system to "cut, copy and paste" elements of job descriptions that cut across the organization of a job description in its textual form.

1. Introduction

The EXCLASS system automates the process of describing and evaluating jobs in the Public Service. EXCLASS uses AI technology to model the reasoning employed by human classification experts in evaluating jobs according to the Universal Job Evaluation Plan (UJEP), a component of the Universal Classification Standard [Treasury Board Secretariat, 1993]. It also incorporates technology for automatic text generation, which permits users to compose and edit job descriptions on a conceptual level, rather than writing ordinary text. Conceptual representations of job descriptions not only provide material for the evaluation component of EXCLASS, but also simplify and standardize the process of creating, modifying, and comparing job descriptions, while freeing the user from some of the mundane aspects of writing textual descriptions.

The two main components of EXCLASS are the Job Description Module (JDM) and the Job Evaluation Module (JEM), which are linked through the User Interface Module (UIM). The JDM, which is the focus of this paper, provides resources for building and modifying conceptual representations of job descriptions, and automatic generation of textual descriptions from these representations. The JEM uses expert reasoning on the conceptual representations to produce a job evaluation[1]. Fig. 1 shows the basic processing model of EXCLASS.

An initial integrated prototype of the EXCLASS system was delivered to Treasury Board in May of 1993. The knowledge base and linguistic data for the prototype are based on analyses of a sample of some 30 existing job descriptions, in the domain of purchasing and supply (comprising subdomains of procurement and asset management). The prototype runs on a 386-class PC under Microsoft Windows™ 3.1; the JDM is implemented in LPA 386-Prolog for Windows.

2. Building conceptual job descriptions

The process of composing and editing conceptual representations of job descriptions is supported by a component of the JDM known as the Job Description Builder (JDB). The JDB allows the user to edit the four types of information which occur in the standardized job description format recently proposed for the public service [Treasury Board Secretariat, 1993]:

- *"Tombstone" Information*—position title, department, location, etc.

---

[1] The JEM (as well as the UIM) was developed by Rockwood Informatics Ltd., Ottawa. For a broader description of EXCLASS, including the JEM, see [Korelsky, Laplante and Forget, 1992].

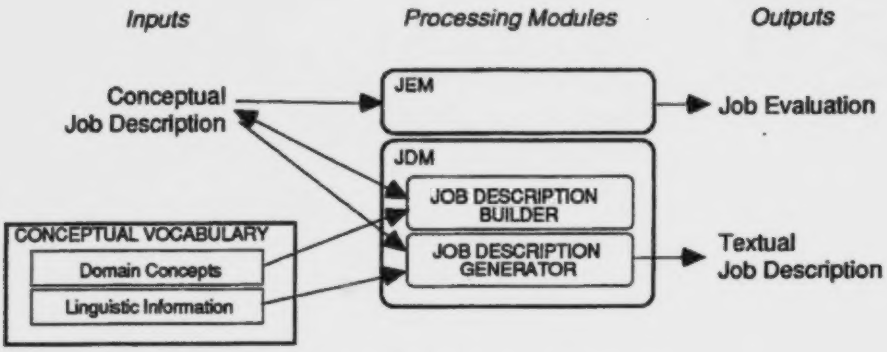


Figure 1: EXCLASS processing model.

- *Client Service Results*—statements of the main results to be achieved by the incumbent. Example: "Warehousing and distribution service to government departments and agencies in the National Capital Region".
- *Key Activities*—statements of typical or important activities involved in achieving Client Service Results. Example: "Planning, organizing and developing warehousing and distribution systems, methods, and techniques".
- *Substantiating Data*—statements about aspects of the job relevant to evaluation, according to the Factors and Elements defined by the Universal Job Evaluation Plan (UJEP). Example (for Element ENVIRONMENT of Factor WORKING CONDITIONS): "The work involves long exposure to an office environment, and occasional exposure to a warehouse environment with large amounts of dust and dirt".

Through the User Interface Module, the JDB provides display and editing functions for conceptual representations of Client Service Results and Key Activity statements. Users compose conceptual forms of these statement types by selecting elements from a conceptual vocabulary, and linking them in attribute-value structures. These conceptual structures are used by the text generator to produce textual forms of the statements, which will appear in the final job description.

The composition of Substantiating Data statements is the same as Client Service Results and Key Activities from the user's perspective; the difference from the system's perspective is that the conceptual elements used for composition reflect the structure of the UJEP, and are manipulated by the JEM component of EXCLASS. The JEM evaluates jobs by comparing Substantiating Data statements to UJEP standards, defined for each Factor and Element.

When it becomes operational, the EXCLASS system will support overall job analysis and editing of tombstone information, through online access to resources such as organizational charts, departmental databases, and program mandates.

3. The conceptual vocabulary

The structured vocabulary of concepts used to compose Client Service Results and Key Activities in the JDM is based on an analysis of existing job descriptions (for the current prototype, in the domain of purchasing and supply). This analysis revealed that some of the useful concepts used to build job descriptions are peculiar to certain domains (e.g. "requests for proposals" for contracting, or "secure storage systems" for warehousing), while other concepts would be useful in a wide variety of domains in the public service (e.g. "budgets" or "client requirements"). The conceptual vocabulary is indexed so that the JDM can refer to different sub-vocabularies of domain-specific concepts, according to the needs of particular users in different departments.

As well as being organized according to domain, the conceptual vocabulary is arranged in a hierarchy of "aspects of service", from most-general to most-specific[2]. The most general concept is "service" itself, which is related to more specific concepts such as "[service] requirements", "activities", "plans", "objects of service" (such as contracts in the procurement domain, or stored materials in warehousing), etc. Examples of concepts which have no more-specific "children" are "the Stores Inventory System (SIS)" and "resource sharing methods". There are also sub-hierarchies, not directly connected to the "service" root, for categories such as "participants in service" (staff,

---

[2] Despite the impression given by Fig. 2, the hierarchy of concepts is non-strict, in the sense that a concept can have more than one parent.

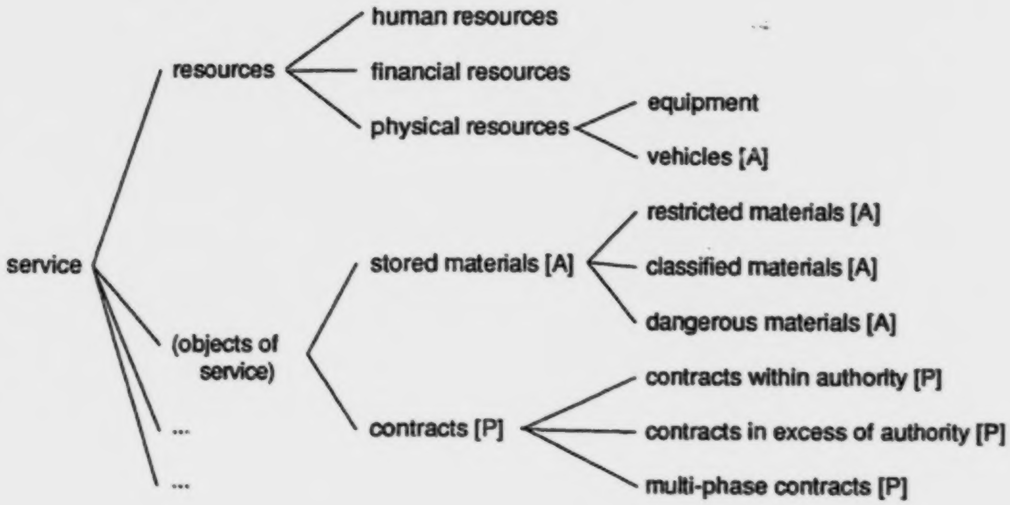
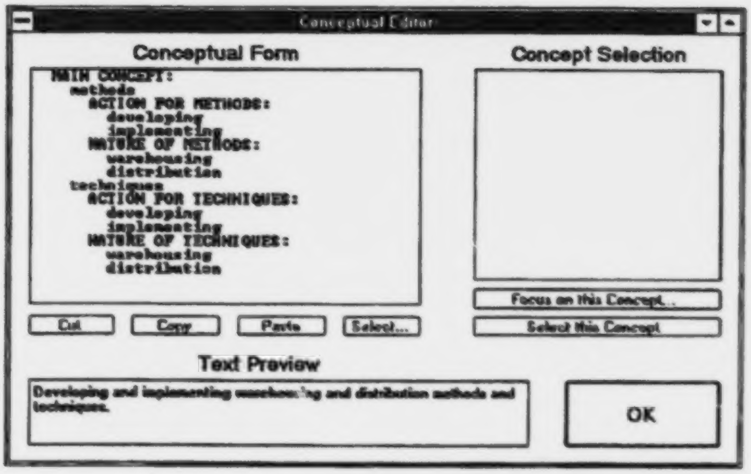

human resources
resources
financial resources
physical resources
equipment
vehicles [A]
restricted materials [A]
service
stored materials [A]
classified materials [A]
dangerous materials [A]
(objects of service)
contracts within authority [P]
...
contracts [P]
contracts in excess of authority [P]
...
multi-phase contracts [P]

Figure 2: Hierarchical organization of the conceptual vocabulary.

clients, government departments, etc.), locations, and "nature of service" (asset management, procurement, scientific procurement, etc.). Fig. 2 shows an example of the hierarchical organization of concepts in the conceptual vocabulary.

The annotations [A] and [P] in Fig. 2 indicate indexing of concepts as being specific to the domains of asset management and procurement, respectively. In order to aid users in selecting concepts from the conceptual vocabulary, the JDM can "switch off" the display of concepts not relevant to the domain in which the user is working. As well, it employs a "focusing" mechanism to display only a small region of the hierarchy at any one time, giving the user means by which to shift the focus up or down the hierarchy to look at more general or more specific concepts, as required.

Each concept from the vocabulary comes with a number (possibly zero) of attributes or slots, which


Conceptual Editor
Conceptual Form
Concept Selection
Focus on this Concept...
Select this Concept
Cut
Copy
Paste
Select...
Text Preview
OK

```
MAIN CONCEPT:
  methods
    ACTION FOR METHODS:
      developing
      implementing
    NATURE OF METHODS:
      warehousing
      distribution
  techniques
    ACTION FOR TECHNIQUES:
      developing
      implementing
    NATURE OF TECHNIQUES:
      warehousing
      distribution
```

Developing and implementing warehousing and distribution methods and techniques.

Figure 3: Conceptual and textual forms of a Key Activity.

can take other concepts as values. It is by assigning values to slots that users construct conceptual representations of job description statements. For example, Fig. 3 shows the conceptual and textual forms of a Key Activity.

In the box labelled "Conceptual Form" in Fig. 3, upper case indicates slot labels, and indentation indicates which concepts are in which slots. When the user selects a slot in which to add a concept, s/he is prompted to select either (1) from a fixed list of possibilities (as in the case of an ACTION slot for any concept), or (2) by browsing through the concept hierarchy (as in the case of NATURE slots, which take their values from the "nature of service" sub-hierarchy, or the "dummy" MAIN CONCEPT slot, for which the user starts at the root concept "service", and can browse anywhere in the main vocabulary).

4. Generating textual job descriptions

Conceptual representations of job description statements abstract from peculiarities of textual job descriptions such as word or phrase order, and ambiguities arising from conjunction. They provide a level on which users can compare, search and "cut and paste" job description statements, while preserving their conceptual coherence. A further advantage of their use in the JDM is that once conceptual forms of job descriptions have been composed, the details of producing finished textual forms, in a standard professional style[3], are handled automatically.

The approach to text generation embodied in the JDM differs from other generation systems, in that most "text planning" tasks are handled by the user, rather than an intelligent component of the generator. The generator mainly performs "text realization" functions – the most obvious of which is the treatment of sometimes complex conjunctions, as in Fig. 3 above. By manipulating the order of elements of the conceptual representation, the user can affect the order of the corresponding conjuncts in the textual form.

In order to facilitate users' manipulation of conceptual representations, and to indicate clearly how these correspond to textual forms of job description statements, the JDM makes use of a novel concept of "continuous text feedback". Rather than the approach initially envisioned, on which the user would compose substantial portions of a job description purely on the conceptual level, then request generation of the corresponding text *en masse*, the JDM employs a design on which every change a user makes to a conceptual form causes a "text preview" box in the conceptual-editing window to be updated (see Fig. 3). With this feature, the user has instant confirmation of the effects that any conceptual change will have on the text of a particular statement, without having to repeatedly invoke the generator. It is anticipated that this feature will also be helpful to users of the system who are initially unaccustomed to composing job descriptions on the conceptual level.

5. Future plans

5.1 Research and Development

Two main directions have been proposed for future research and development, in order to prepare the Job Description Module of EXCLASS for incorporation into an operational system: bilingual generation of job descriptions, and improvements to the knowledge representation scheme used by the JDM.

Given that the JDM already works with "deep" conceptual representations of job descriptions, it is straightforward in principle to modify the text generator so that it can produce textual descriptions in French as well as English, thus adding to the utility of the EXCLASS system. The main research question will be how much conceptual representations need to be modified, if at all, in order to represent a "language-neutral" view of the concepts underlying job descriptions. It may be that English and French speakers see these concepts in subtly different ways, which are influenced by differences in the languages which they speak.

The key to the usefulness of the JDM component of EXCLASS is the adequacy of its conceptual vocabulary for describing jobs in various domains in the public service; therefore, most of the work in preparing it for eventual deployment in government departments will consist in expanding and improving the conceptual vocabulary, chiefly by developing concepts specific to new occupational domains. Research will also be needed into whether the current knowledge representation scheme will scale up adequately to a greatly increased volume of concepts.

In order to aid in the expansion and maintenance of the conceptual vocabulary, special software tools will be designed to support an "acquisition" function. This will allow trained, designated personnel to easily add new concepts to the vocabulary, by modifying "templates" based on existing concept types. In this way, users of the system will have a direct means of incorporating their own expertise into it.

---

[3] The text produced by the JDM represents a standardization, and in some cases, a simplification of the human-produced texts on which its output was modelled. Writers of job descriptions sometimes introduce ambiguities or extra information which is difficult to conceptualize consistently.

5.2 Fielding

A plan for fielding the EXCLASS system has been proposed by the developers, comprising three stages: (1) site testing, (2) operational development, and (3) broad deployment.

Site testing involves a group of end users (managers) and systems support personnel in an appropriate government organization, who test and evaluate a prototype of the system, with the conceptual vocabulary modified to suit their particular needs. They provide the developers with feedback concerning any problems noted with the system, and suggested improvements.

Operational development involves taking the system from being an advanced, tested prototype to being a fully operational system. Users at a particular site will use the system on a regular basis, enlarging the library of job descriptions substantially, and make plans for deploying it on a larger scale.

Broad deployment involves making EXCLASS the standard tool for job description and evaluation throughout a government organization. At this stage the developers will address needs and opportunities arising from a broader range of users, and an enlarging base of job descriptions – for example, the possibility of indexing collections of conceptual job descriptions, or producing different types of textual output for different users of the system.

References

Korelsky, T., Laplante, M. and Forget, S., [1992] "The EXCLASS Job Description and Job Evaluation System", in J. Brahan and G. Lasker (eds.), *Proceedings of the 6th International Conference on Systems Research, Informatics and Cybernetics (Baden-Baden)*, vol.2, pp. 65- 72.

Treasury Board Secretariat [1993], *Universal Classification Standard (Draft)*, Treasury Board of Canada, Ottawa, Spring 1993.

905C20

Lessons Learned Using an Expert System Shell[1]

J.R. Amyot and W. Boev
Knowledge Systems Laboratory, Institute for Information Technology
National Research Council Canada, Ottawa, K1A 0R6

#149709

Abstract

Based on experience with the use of Nexpert Object 2.0B (an expert system shell from Neuron Data, Inc.) in the development of diagnostic applications, principles and practices for good performance were formulated. Some of these approaches are specific to this shell and others are general to similar types of shells. Their application can improve the response time by a factor of 10 to 100, typically. This paper describes the authors' experience with the implemenation of a specific diagnostic method using this shell and presents the aforementioned principles and practices. It gives some insight into the capabilities and limitations of this well-known, relatively high-end shell with portability across several platforms (e.g., SUN and DEC workstations, and PCs) that has a companion GUI builder (Open Interface) with portability across several windowing systems (Microsoft WINDOWS, Motif, Open Look, Presentation Manager and MacIntosh). It also shows how to overcome some limitations that are specific to this shell, and others that are common to shells with similar object-oriented features.

Introduction

The authors have used Nexpert Object 2.0B (an expert system shell) and Open Interface 2.0 [the companion GUI (graphical user interface) builder] from Neuron Data, Inc. fairly extensively in the development of GENDIAG (a generic framework for diagnosis) and in the application of GENDIAG to a pulp bleaching diagnostic expert system. The objective of this paper is to describe problems encountered in the use of this software and methods employed in their solution.

The pertinence of this information is due to: (a) the relative popularity of this software, based on its functionality and wide portability to some 35 platforms (e.g., SUN and DEC workstations, and PCs) and 5 windowing systems (WINDOWS, Open Look, Motif, MacIntosh, and Presentation Manager), and (b) a degree of commonality between this shell and other object oriented shells of this type. New versions of the targeted software currently being marketed are removing some (but not all) of the limitations of their predecessors. Even where limitations are removed, the insight gained from the experience presented here may be useful in troubleshooting.

Application

Nexpert Object [1] was used to develop GENDIAG, a generic framework for diagnosis, consisting of a diagnostic inference engine and a user interface. GENDIAG, in turn, was used to develop a diagnostic expert system for the chlorination stage of bleaching in a softwood Kraft pulp plant.

The function of GENDIAG is to identify and rank the possible faults associated with a set of known symptoms and to determine appropriate actions to rectify the faults. Symptom parameter values (e.g., TRUE, FALSE, HIGH, DECREASING, NOTKNOWN, etc.) are specified by the user through the medium of four forms. From these values, GENDIAG produces a fault ranking in descending order of likelihood. The user can specify more parameter values (or change the existing ones) to improve the diagnosis, or he can take some remedial action to correct the fault(s). This is an iterative process that can be terminated at the user's discretion. The pulp bleaching application's knowledge base consisted of 33 faults, 36 symptoms and approximately 250 rules relating the faults to the symptoms.

Computational Environment

The following is a list of the tools that were used during the development of GENDIAG:

(a) Nexpert Object 2.0B from Neuron Data, Inc.
(b) Microsoft C/C++ 7.0
(c) WINDOWS 3.1

---

[1] NRC 37049

(d) DOS 5.0 and 6.0
(e) WINDOWS 3.1 SDK
(f) Open Interface 2.0 from Neuron Data, Inc.
(g) Borland C/C++ 3.0

Some tools, such as Borland C, were used only temporarily. Open Interface [2] was used to improve the user interface. Microsoft C/C++ was chosen over Borland C/C++ since Neuron Data supports Microsoft products more completely than Borland products. With Microsoft C/C++, it is possible to use Nexpert C library routines to query the Nexpert knowledge base. With Borland C/C++, it would be necessary to rewrite those routines.

Nexpert Object Overview

Some features of Nexpert Object include:

(a) object oriented
(b) pattern matching capability
(c) several inference strategies
(d) portability to over 35 platforms
(e) character based user interface plus GUI capability
(f) application programming interface (API) for integration with other programs

Nexpert Applications under DOS

Nexpert Object has a character based user interface (NXPFORMS) which can run in various character based computing environments including DOS. There are several versions of NXPFORMS ,which are needed depending on the size of the application and the configuration of the hardware (Figure 1). For instance, the virtual memory versions are needed if the application environment does not have enough extended memory, while the large versions are needed if the knowledge base exceeds 10000 objects (approximately). When using the virtual memory versions, it is also necessary to copy the NXPMSVR.VMC file to the NEXPERT directory.

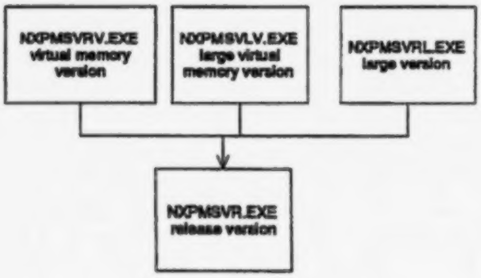


FIGURE 1: Nexpert under DOS.

NEXPERT Applications under WINDOWS

There are two main options for developing Nexpert applications under WINDOWS:

(i) Front-end DLL (release version) or Front-end DLL (debugging version) (Figure 2)
(ii) Embedded DLL (Figure 3)

A DLL is a dynamic link library which is linked at run time.

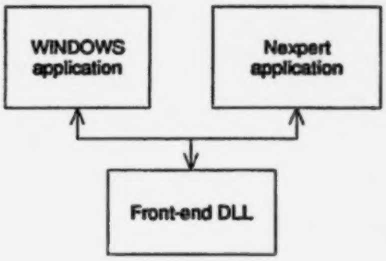


FIGURE 2: Front-end DLL.

With the first option, the development system runs concurrently with the application. One can switch to the development environment and run the Nexpert editors, networks, etc. Release and debugging versions are available under this option. The debugging version writes messages to a file called NXPW.DBG; hence, its execution can be slow. The release version is the default version.

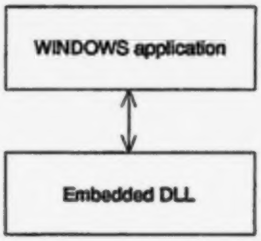


FIGURE 3: Embedded DLL.

The second option (embedded DLL) was used for the WINDOWS version of GENDIAG. This DLL contains the entire functionality of Nexpert Object, but without an interface development facility. This was provided by Open Interface, another Neuron Data product. The embedded DLL (release version) runs 15% faster than the front-end DLL.

The different versions can be run by overwriting the file NXPW.DLL with the files shown in Figure 4.

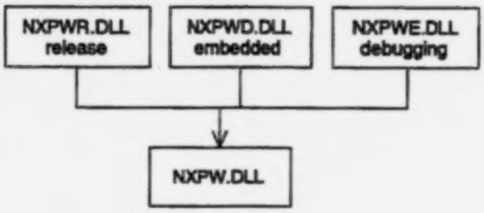


FIGURE 4: Nexpert under WINDOWS.

NEXPERT API Applications under DOS

Under DOS, the Nexpert kernel is available as a server. Communication is provided by a special DOS library (NXPMCLI), which handles transparently the links between Nexpert and the application.

NEXPERT API Applications under WINDOWS

Since we are linking with a DLL (dynamic link library) certain rules have to be observed when setting handlers (i.e., when calling outside routines from Nexpert or having an outside routine call Nexpert). Under WINDOWS the DLL is called NXPW.

Setting Handlers

All outside routines written for a Nexpert WINDOWS application must follow WINDOWS conventions for setting handlers for the routines. The most important ones are:

(1) "export your handler functions in your .DEF file. As the functions are C instead of PASCAL, you will need to prefix them with an underscore." (example .DEF file provided in Appendix B)
(2) call the WINDOWS API function MakeProcInstance and pass the result to NXP_SetHandler or NXP_SetHandler2 (example provided with the call to the gendiag() routine in Appendix A).

Performance

GENDIAG 1.0 was developed under a tight schedule, without prior experience in the use of Nexpert, and with a strong emphasis on functionality relative to the pulp bleaching application. This resulted in performance issues which were addressed in version 1.1. Some of the factors leading to poor performance are:

(a) knowledge base size
(b) calling rules which are not needed
(c) object search domain not suitably restricted
(d) using an improper inference strategy
(e) overuse of the RESET operator
(f) unnecessary (time consuming) creation of dynamic objects
(g) string manipulation (more efficiently done with C/C++)

The pulp bleaching application took over one hour to perform a diagnosis with GENDIAG 1.0. This execution time was reduced to less than 30 seconds with GENDIAG 1.1. This latter version had the same functionality as the previous version. Only the performance was improved - by methods based on the above factors, as described below.

Inference Strategy Setting

Nexpert Object 2.0B has four types of inference strategy: (1) backward chaining, (2) forward chaining, (3) gating, and (4) context links. All these strategies are nominally enabled by default, but one or more of them can be disabled.

However, all the strategies were left enabled in version 1.0 of GENDIAG, even though only backward and forward chaining were used. The context links strategy was inconsequential since a context link must be declared explicitly in a rule. On the other hand, gating provided quite a bit of overhead; Nexpert was repeatedly placing hypotheses on the agenda, analyzing them and eventually rejecting them since they were not part of GENDIAG's strategy. A substantial reduction in execution time (over 60%) was achieved by disabling the Nexpert gating mechanism.

Use of the RESET operator

It is possible to implement loops in Nexpert Object through the use of the RESET operator. A hypothesis is a rule with a name that has a boolean or "unknown" value. When the rule's conditions evaluate to TRUE, the hypothesis is set to TRUE and the rule's actions are executed. If one of the actions is to reset the hypothesis (i.e., to unknown), then the hypothesis is placed back on the agenda to be executed again. This process is repeated in a loop that terminates when the rule's conditions do not all evaluate to TRUE. A reset action automatically resets all hypotheses that are executed as subgoals in the rule's conditions. Therefore, it is wasteful of execution time to reset the subgoal hypotheses independently when they are already automatically reset.

Creation of Dynamic Objects

Creating dynamic (run-time) objects consumes significant execution time with Nexpert. Therefore, its use and the number of objects created should be limited.

Calling Mutually Exclusive Hypotheses

A Nexpert hypothesis consists of 3 elements:

(1) left hand side conditions (all of which must evaluate to TRUE in order for the hypothesis to be set to TRUE),
(2) hypothesis name which can be TRUE, FALSE or UNKNOWN, and
(3) right hand side actions which are executed if the hypothesis is TRUE.

(i.e., conditions --> hypothesis --> actions)

When a hypothesis is called, it is placed on the agenda if its value is UNKNOWN. Thus, a hypothesis H2 can be evaluated with the (pseudo code) statement

```
Call    H2
```

If H2 is Unknown, Nexpert will evaluate its left hand side conditions to try to determine if it is TRUE or FALSE and, if necessary, will even backward chain on any hypotheses that lead to it (subgoals). Should H2 be initially TRUE, then it will not be placed on the agenda since its value is already known.

Suppose we have a hypothesis H1 which calls two other hypotheses (H2, H3), i.e., H1 has the following statements in its LHS conditions (all hypotheses initially RESET):

```
Call    H2
Call    H3
```

Suppose that H2 and H3 are mutually exclusive, such that if one is TRUE, then it is not necessary to call the other. The task is how to get only one of the two rules to be placed on the agenda. One way is to place a test condition on some field in the LHS of H2 and H3 such that when the hypotheses are executed the condition becomes TRUE for only one of the two hypotheses. Thus, even though both hypotheses are placed on the agenda, only one performs any real work related to the application. This, however, reduces the performance of the application since both hypotheses are placed on the agenda while only one is required. Placing hypotheses on the agenda which by their very nature are immediately rejected is very time consuming in Nexpert.

A better solution is to implement a design whereby the mutually exclusive hypothesis is not placed on the agenda. This can be achieved by setting mutually exclusive hypotheses to TRUE at the point where it is determined which hypothesis is currently needed. Consider the following examples in which hypothesis H1 calls two mutually exclusive hypotheses H2 and H3.

```
Hypothesis H1:
        <Conditions>    H1      <Actions>
        Call H2
        Call H3
Hypothesis H2:
        <Conditions>    H2      Set H3 to TRUE
                                <Other Actions>
Hypothesis H3:
        <Conditions>    H3      Set H2 to TRUE
                                <Other actions>
```

The statement

```
Call    H3
```

in hypothesis H1 is executed but, since H3 is set to TRUE by hypothesis H2, it is not placed on the agenda for evaluation. It is also imperative that the hypothesis most likely to be executed be placed first, since it will provide a greater chance of eliminating the hypothesis below it.

Limiting the Object Search Domain

Rules, in Nexpert, can select a certain subset of the objects using a technique called pattern matching. The properties of these objects can then be manipulated or changed using a single command which operates on all the objects selected. The selected objects are local to the rule and if we need to operate on the same subset of objects in another rule we must do another pattern match to once again select the proper subset. This is time consuming since Nexpert goes through all the objects to select the few with the desired attributes. It would be much better, if once having selected a certain set of objects, we could then manipulate this set in as many rules as might be desired without having to perform redundant pattern matchings. Nexpert does not provide a built-in technique to achieve this, so we had to devise a method.

Let O1C, O2C, O3C, ... be objects of class C. Suppose we were to do a pattern match on the class C to select the object O2C. The pattern match is represented by <C> (i.e., <C> represents the object O2C). Let there be associated with the object O2C a set of other objects O2R1, O2R2, O2R3 and O2R4, all of class R (Figure 5).Suppose class R consists of many other objects, but we are currently only interested in these four since they share a common property, say O2. We can isolate these four objects by doing a pattern match on all the objects in class R that have the property O2. However, we cannot transfer this pattern match directly to another rule since the objects are local to the current rule.

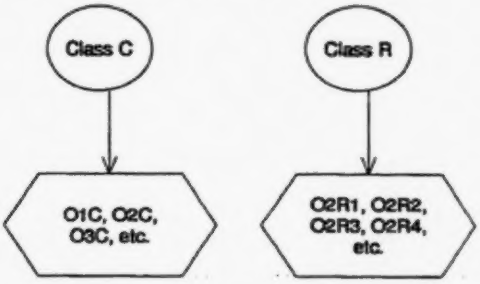


FIGURE 5: Objects of classes C and R.

Our technique is based on the following reasoning. In Nexpert, a single object can be a subobject of more than one class or object. Hence if we were to declare the objects O2R1, O2R2, O2R3 and O2R4 to be subobjects of the object O2C we could, through the use of some simple coding techniques, transfer the objects to another rule and not have to perform any extra pattern matching (Figure 6).

The transfer technique is as follows.

(a) Make sure that each object has a string property, say "name", which takes a value equivalent to the object name (e.g., O2C.name has a value "O2C").

(b) O2R1, O2R2, O2R3 and O2R4 are declared as subobjects of O2C. (Note that this makes the objects O2R1, O2R2, etc. objects of both class C and class R).

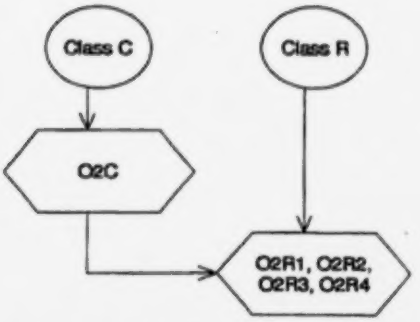


FIGURE 6: Objects with multiple associations.

(c) An object, say t, is created with a single string property, say x.

(d) Transfer the name of the pattern matched object to t.x. (In Nexpert this is done with the following piece of code: Name <|C|>.name t.x). At this point, we can do a Nexpert interpretation on t.x to select the object O2C. This is done through the statement \t.x\ (i.e., \t.x\ gives us O2C).

(e) In another rule, do a pattern match on the Nexpert interpretation \t.x\ to once again select the objects O2R1, O2R2, O2R3 and O2R4. Thus the Nexpert statement <\t.x\> would once again select the desired objects without going through all the objects of class R to make the selection.

String Manipulation

Nexpert's sophisticated API makes it possible to perform certain functions using outside routines in a language more suited to these functions. Custom C-language functions for string manipulation were more efficient than the equivalent Nexpert functions.

Knowledge Base Size

The size of the knowledge base is a factor in reducing execution speed. In our experience, it starts to show its influence when the total number of objects reaches approximately 500 to 1000. A technique which might be useful is to divide the knowledge base into several knowledge bases and load only the ones that are currently relevant. Any time saved by performing the inferencing on a smaller knowledge base has to be balanced against the time it takes to load and unload the knowledge base modules as they are needed.

Integration

Version 1.1 of GENDIAG is a DOS application with a character based user interface (CBUI) developed with NXPFORMS, the user interface development portion of Nexpert. This version was converted to a WINDOWS application (version 2.0) with a graphical user interface (GUI, including buttons, icons, multiple windows, etc.) developed with Neuron Data's Open Interface 2.0. This section describes some of the problems and solutions pertaining to the conversion from DOS to WINDOWS. The problems revolve around the integration of: (a) WINDOWS routines, (b) custom C-language routines, and (c) GUTs developed with Open Interface 2.0.

WINDOWS Routines

Neuron Data has provided template files which, when properly used, convert DOS programs to WINDOWS. The most important of these is whandler.c. This template "can convert call back handlers which don't utilize any keyboard I/O (typically computation or file access routines). The file whandler.c takes care of creating a WINDOWS application which will handle communication with WINDOWS and Nexpert". To convert a DOS routine, the whandler.h file must be included in all C files and the whandler.def must be added to the handlers. The routine is then compiled and linked with the appropriate flags to produce an equivalent WINDOWS program. Sample makefiles are provided in the documentation and are run with the command: nmake -f makefile.win.

Two problems were encountered when running the program produced as described above:

(1) An error occurred because there is more than one main window in the application. The procedure described above links to an object file whandler.obj. It is necessary to edit the file whandler.h to add the line

```
#define main mymain
```

and create a new object file whandler.obj.This problem did not occur with Microsoft C 6.0 but did with version 7.0 .

(2) In some circumstances, our executable file did not work unless it was also linked with a library called oldnames.lib found in the Microsoft C compiler libraries. This problem might be related to the fact that version 7.0 of the Microsoft compliler was being used. It is recommended that this file be included in all links.

Custom C-language Routines

One way to call outside routines is to embed them in the application using NXP_SetHandler or NXP_SetHandler2 and run it as one process. The Nexpert API libraries can then be used to query the knowledge base (Figure 7).

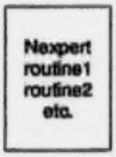


FIGURE 7: Nexpert running as one executable with outside routines.

Another way is to spawn a separate process (independent of the Nexpert API) which will handle certain files output by Nexpert, change the files and have Nexpert eventually read these files back in. With this method, the spawned process runs, uses memory and releases it to the application. Thus it is conceptually possible to run much larger applications. With the first method the whole application runs as one executable file and might run up against the DOS 640K limit. Figure 8 illustrates the second method.

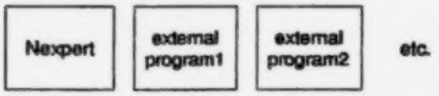


FIGURE 8: Three or more separate executables.

Unfortunately (as of the time of writing) only the first method is possible under WINDOWS. With the second method, the spawned process would run once and never run again until we exited WINDOWS and came back in. The reason is that the spawned process remained in memory and WINDOWS could not load more than one copy of the process at a given time. Neuron Data's initial recommendation was to use only the first method with WINDOWS applications. A later suggestion was not pursued.

Open Interface

Both Open Interface 2.0 and Nexpert Object 2.0B require a WinMain() as an entry point. This is a basic requirement of all WINDOWS applications. Since there can only be one WinMain() in an application either one or the other WinMain() has to be renamed and rewritten. Nexpert, when running under WINDOWS, needs to know the WinMain() routine parameters. Therefore it is necessary to save these parameters when Open Interface starts up and to make them available to Nexpert when the need arises.

Open Interface does not give easy access to its WinMain() routine. When an interface is built using Open Interface a certain number of C files are generated only one of which has a main() routine. Note that this is not a WinMain() routine but rather simply a main entry point to the interface. When an Open Interface application is compiled, linked and run, a number of things occur which are hidden from the software designer. Any WINDOWS application demands a WinMain() as an entry point. So Open Interface, (when it starts up) calls a WinMain() routine which is in a file called app.obj. This object file, in turn, calls the main() routine of the Open Interface application. Since the Open Interface application has a main routine called main(), it is necessary to rename this routine to something else; otherwise, an error will occur. Thus Open Interface contains the following line in the file apppub.h:

```
#define main APP_Main
```

This is necessary since the presence of a main() routine (even though it is not WinMain()) will cause the Microsoft compiler to ignore the WinMain() routine. (NOTE: The app.obj file is obtained by compiling the files app.c and mswapp.c in OPENINT\MISC).

When designing a GUI with Open Interface, it is recommended that the whole interface be contained in a single library. Problems were encountered when two people designed parts of the interface in two separate libraries. It is possible to combine separate libraries, but there is no built-in method of doing it. Hence if the interface is contained in more than one library it is necessary to manually modify the resource files, keeping track of proper names, etc. After modifying the files it is necessary to use the resource compiler (RESCOMP) to produce a new application. This method proved to be time consuming and prone to errors. It is therefore recommended that the entire interface be designed as one library.

Since a WINDOWS application cannot run under the huge model it is necessary that there be no segments which take up more than 64K of memory. If an error occurs stating that a segment is more than 64K, it may be necessary to reduce the stacksize and set the Gt switch, which will ensure that segments are created correctly in a 16 bit computer. This problem was encountered with GENDIAG and solved by reducing the STACKSIZE from 24576 down to 20000 and using the Gt switch when compiling.

Discussion

Nexpert provides database bridges to several databases as an easy way of entering data into the Nexpert knowledge base. In some instances it may not be possible to use these database bridges (e.g., lack of database software or lack of time to learn to use a certain database). In such instances, Nexpert provides graphical editors, networks, etc. through which one could enter data. However, these are quite cumbersome to use when entering large amounts of data. For instance, when initializing a property of an object, one must click on several buttons in different windows in order to get to the point where one can initialize the property. In such circumstances, it is recommended that the property values be read (at the beginning of the application) from one of the two ASCII file formats allowed in Nexpert, i.e., NXP or NXPDB. Note that this concerns entering data into a knowledge base in the absence of a maintenance interface (e.g., when designing quick prototypes).

Open Interface gives complete portability over several GUI platforms. It maintains portability across these platforms by providing a Virtual Graphics Machine which gives a superset of all the widgets available in each one of the native GUIs. This superset is implemented separately from the native toolkits. By including the native toolkits' library files we can mix Open Interface calls with calls to the native toolkit. However this should be avoided at all costs if the desired result is portability. Even a single call to a native function will make the application no longer portable.

The massive use of dynamic objects should be avoided unless done only once as in the beginning of a program when reading in the object properties from a file.

Nexpert is a relatively high-end shell. Many of the techniques advocated here to improve its performance should be applicable to object oriented shells. The new version of Nexpert Object (Smart Elements), according to Neuron Data documentation, is a fully object-oriented expert system shell with a graphical user interface. The GUI is taken from elements of Open Interface. This improved version provides better message passing facilities among objects, generates C++ code instead of C code, and lets the user specify both system and user-defined methods. User defined methods are implemented explicitly according to application design while system defined methods are implemented transparently to the user by Nexpert.

Conclusion

The authors' experience with Nexpert Object 2.0B and Open Interface 2.0 (both Neuron Data, Inc. products) in diagnostic applications was described. Several methods to improve performance (i.e., reduce execution time) were presented. Some are pertinent to similar types of object oriented shells. Several difficulties had to be overcome when converting an application from DOS to WINDOWS, and when integrating an Open Interface GUI with a WINDOWS-based Nexpert application.

The difficulties associated with the latter integration have been eliminated in Smart Elements, the new product that integrates Nexpert Object 3.0 and Open Interface 3.0. Some performance improvement is also claimed for this latest version. Special care has been taken to ensure that applications with thousands of objects perform adequately. The previous versions gave very good functionality and excellent portability, but care had to be exercised to achieve good performance and integration. A new feature, Open Script Element, has been added. This feature links interface widgets to Nexpert objects and describes their behavior on screen. With the recent improvements, the authors feel that the new product should continue to be a strong contender in the field of expert system shells.

References

1. Neuron Data, Inc., "Nexpert Object 2.0B manuals."
2. Neuron Data, Inc.,"Open Interface 2.0 manuals."

Appendix A: Calling Outside Nexpert WINDOWS Routines under Open Interface

In order to access all the parameters of the WinMain function it may be necessary to declare several global variables which can then be set to the WinMain parameters and then used in other parts of the interface. The WinMain routine is found in the file mswapp.c. Thus we might declare the following four new variables in mswapp.c

```
HANDLE          whInstance;
HANDLE          whPrevInstance;
LPSTR           wlpCmdLine;
int             wnCmdShow;
```

Then in the WinMain routine we can gain access to the parameters in the following manner:

```
int PASCAL WinMain(hInstance, hPrevInstance,
lpCmdLine, nCmdShow)
HANDLE          hInstance;
HANDLE          hPrevInstance;
LPSTR           lpCmdLine;
int             nCmdShow;
{
whInstance      = hInstance ;
whPrevInstance  = hPrevInstance ;
wlpCmdLine      = lpCmdLine ;
wnCmdShow       = nCmdShow ;
}
```

These new lines make available to Nexpert Object certain handles which may be necessary. The new app.obj is produced by linking in the new mswapp.obj file thus produced.

Having seized control of certain handles needed by Nexpert Object we can now interface with the routines which load the Nexpert application. We use NXP_SetHandler to name the outside routines called by the Nexpert Object application, and then run it.

Declare as extern the variables which are obtained from the WINDOWS WinMain routine.

```
/* The Open Interface function load_NXP loads and
runs a Nexpert knowledge base. We thus have the
following line in the Open Interface code */
          load_NXP( whInstance ) ;
```

The actual function load_NXP would look as follows

```
/*gen.tkb is a Nexpert Object application    */
/* which calls an external C function "GENDIAG" */
int   load_NXP(S_hInst )
HANDLE S_hInst ;
{
KBId   testKB;
AtomId  hypo ;
FARPROC      gen;

FARPROC  instProc;
int  success,  load ;

NXP_Control(NXP_CTRL_INIT);
/* set handlers */
success=NXP_SetHandler(NXP_PROC_EXECUTE,
(NxpIProc)MakeProcInstance((FARPROC)GENDIAG,
                S_hInst), "GENDIAG");
load = NXP_LoadKB("gen.tkb",&testKB) ;
NXP_Control( NXP_CTRL_RESTART ) ;
NXP_GetAtomId("gen",&hypo,NXP_ATYPE_SLOT);
NXP_Suggest(hypo,NXP_SPRIO_SUG ) ;
NXP_Control( NXP_CTRL_KNOWCESS ) ;
NXP_Control(NXP_CTRL_EXIT);
return 1 ;
}
```

Appendix B: Sample DEF file

The routine gendiag() is called from a Nexpert Object application and hence it must be placed as an EXPORT in this file. This file is linked in with the application.

```
NAME  Ex1
...
/* Standard .DEF file statements    */
...
EXPORTS
      _gendiag
```

905C4C

Knowledge-Based Systems for Managing Complexity

A Report from the Special Panel on KBS

Erik Solem[1]
Directorate of Social and Economic Analysis
National Defence Headquarters,
Ottawa, Ontario K1A 0K2

#149710

Background

For the 1993 DND Workshop on Advanced Technologies in Knowledge-Based Systems and Robotics, Ottawa, 14-17 November 1993, a special panel entitled **Knowledge-Based Systems for Managing Complexity** was conceived and chaired by the above author, with the following additional panel members-participants:

Mr. Stan Isbrandt - Emergency Preparedness Canada
MGen (Ret'd) George MacFarlane
Col Conrad Mialkowski, Assoc. DGRD Ops, CRAD
Dr. Zbigniew Mikolajuk, IDRC
Mr. Don Smith, DREV, and
Mr. Grant Thomas, RES Policy Research.

Messrs Isbrandt, MacFarlane, Mialkowski and Smith had already participated in two previous panels, on the "Applications of Expert Systems in DND, Present and Future" under the same chairman/moderator as in this case. Hence, for them it was very much a "follow-on" workshop with the testing of hypotheses already presented and the introduction of new findings and ideas based on the foregoing panel discussions. Dr. Mikolajuk and Mr. Thomas, both novices to this particular group, had prepared texts, submitted for publication, prior to having been selected for the KBS panel, hence their papers may be found elsewhere in this set of Proceedings.

Introduction

As has already been fully established within the context of this panel, the military uses of Knowledge-Based Systems (KBS) or Expert Systems are many and varied.[2] During the First Workshop on Applications of Expert Systems in DND, held at RMC, where the present panel was initiallly formed, many of the subsequent ideas were first introduced and discussed. Likewise, the initial conceptional modelling efforts were put forward within the context of that panel. We looked then, as now, at military as well as civilian uses of these systems and processes which make up KBS applications, since in the world of advanced technologies the civilian and military uses often blend. As anticipated then, developments and applications areas which may have been considered to be at the "outer edges" of the possible or even desirable, are now seen to have become imbedded in the general way of doing things. Those with a particular interest in following the evolution of our concepts and ideas are encouraged to read the proceedings of these early ES Workshops/Symposia.

---

[1] Phone (613) 992-4541

[2] See, for example "Military Uses of Expert Systems: Some Future Perspectives" by Erik Solem, presented to the Symposium/Workshop on Applications of Expert Systems in DND, 2-3 March 1989, Royal Military College of Canada, Kingston, Ontario, published in its Proceedings.

Findings and Results

Turning to the present panel, the aim of **Mr. Isbrandt's** particular contribution this time was to illustrate how an increasing level of complexity in computer applications can be handled with Knowledge Based Systems.

Coming from an Operational Research perspective, the particular focus was to discuss how Knowledge Based Systems can assist with the development and maintenance of a mathematical simulation. The discussion was based on lessons learned in the Directorate of Manpower Analysis (D Man A) in DND.

Several years ago, manpower modelling programs which were in use were relatively inflexible. The addition of one new feature to a model would often require changes throughout the application, requiring months to incorporate. New projects coming on line needed a much more flexible approach, in order to be able to incorporate needed changes within a reasonable time frame. This led to the development of a much more flexible model building utility, called GeM.

GeM has been successfully used as the basis for several different manpower models. It allows the building of complex mathematical simulation models from an object-oriented, menu-based interface. Model elements and dynamics are represented using objects such as nodes (which represent states) and links (which represent transformations of a resource from one state to another). Since standardized menus cannot foresee all eventualities, GeM allows the incorporation of a distinct Knowledge Based System in any link. This allows the model builder a high degree of freedom above and beyond the facilities provided by the GeM menus. A complex model may use a number of independent, compartmentalized Knowledge Based, or Expert Systems, in order to achieve a more accurate representation of real dynamic processes.

GeM has been successfully used to create several different models for manpower analysis. A typical model begins with a starting snapshot of an occupation, and projects 20 years or so, year by year, for an occupation of hundreds or thousands of individuals.

According to Isbrandt, the groundwork has been laid to continue on with functional models, where platform operations would be modelled in conjunction with occupational career progression and training. For instance, it would be feasible to create a model where operational cycles for naval platforms (duty periods, refit periods, etc) were modelled over a long period of time in conjunction with career progression, training, and sea-shore rotation for occupations which man the platforms. Such extra complexity is handled by incorporating individual sub-models in a larger GeM model, and then adding the interactions between sub-models using imbedded Knowledge Based Systems as necessary. Thus, more complex models are handled by aggregating sub-models, and modelling the interactions.

In practice, an analyst might run and rerun a functional model as described above, continually changing model parameters until model behaviour exhibits some desired feature, or until the best result (measured with some yardstick of optimality) is obtained. Instead of an analyst choosing how to alter parameter values in this quest for an answer, it makes sense to use a computer, with a machine learning module, to change values and rerun the model. In D Man A a simple process of this type using a Genetic Algorithm has been tested as a proof of concept prototype.

Basically, this illustrates one process whereby Knowledge Based Systems can be used to handle increasing levels of complexity. GeM, as a model building utility, incorporating KBSs, has been used to create increasingly complex models which incorporate complexity which was impractical to consider previously. Isbrandt argued that the limits of GeM's ability to handle complexity have not yet been reached, so it can be used to create aggregate models of still more complexity. By incorporating GeM models within a machine learning loop, the computer may also be able to relieve the analyst of the complexity inherent in an iterative process of finding better or optimal model parameters.

In a generic sense, according to Isbrandt, the same approach can be used in other complex application areas. Sub-systems can be modelled independently, using their own Knowledge Based Systems or Artificial Intelligence features as appropriate. By providing a framework which is designed to accept and integrate generic modules, larger systems can be created. These aggregations may occur over several levels, and the interactions among modules may need to incorporate their own KBSs or other AI features. However, by designing the framework from the beginning to be amenable to a "mix and match" mode of combining and aggregating sub-systems, great flexibility can be attained in the handling of increasing levels of complexity.

MGen (Ret'd) George MacFarlane, a graduate of UBC (1954) and of the University of Michigan (1962), a former RCAF officer having retired as MGen, and subsequently Associate ADM(Mat), was most recently Executive Director of the Institute for Space and Terrestrial Sciences (ISTS). MacFarlane spoke on "Enabling Technologies".

The issues in the Information Age (which may be defined as instant communications, accelerated pace of invention and discovery, and rapid diffusion of knowledge) is that with a technological explosion there is simultaneously a population explosion (1990 - 5.4 Billion, 2030 - 10 Billion), with subsequent pollution and toxic waste. Global change itself presents problems of climatic change and more frequent, more severe storms; rising mean sea level; and implications for agriculture, forestry, fisheries, water supply *and for society itself.*

Discoveries and inventions build upon previous ones. However, in the pre-information age knowledge dissemination was slow, the discovery and invention rate sporadic and low and there was ample time for society to adjust itself. Not so with the coming of the information age. From now on, dissemination and diffusion will be **rapid**, discovery and invention rate **high** and **accelerating**, while the adjustment time only a few years to months.

MacFarlane then traced the development of the AI/Robotics/KBS discipline and gave examples of representative systems. These were Medical Diagnostics, e.g. MYCIN, EMYCIN. Decision Support and C^3I, self-paced Individual Student tutoring, On-condition Monitoring and Maintenance Advice, and others.

From this he moved to implications of Post-2000 demographics and technological advances. With fewer entrants to the (military) labor force there would be greater competition; the need for greater retention with subsequent greater job satisfaction required, and perhaps even a reduction in requirements.

The flood of new technologies, with new capabilities and faster diffusion worldwide should be seen in juxtaposition with increased likelihood of global unrest. However, our defence now really means few weapons systems, few war reserves and a weak defence industrial basis.

Key steps to at least partially counter these developments should be to select promising applications (which should be manpower intensive and reaction time critical). Secondly, we must try to convert theory to practice. This would, and should, be done by acquiring competence with regard to design, manufacturing, operations, maintenance and management. It would also mean acquiring understanding of the potential of new technologies. MacFarlane saw several promising military applications of technology, which he outlined.

The thrice-serving member of the panel, **Colonel Conrad Mialkowski** offered a presentation which he had entitled "Managing Complexity - A Soldier's Glimpse". Mialkowski, who had joined the Army in 1957 with two terms in Germany and three terms in NDHQ has been the Project Director of TCCCS and is presently Associate DGRD Ops of CRAD.

In a very useful overview, he surveyed the environmental impacts since the last conference. These were: continued world unrest, the increasing role of peacekeeping with what that entails; changing threats and (new) Canadian concerns, e.g. employment, debt and regionalization. This is all taking place within a situation of continued economic pressure, with fewer defence resources (people and money) and a decreasing national will for defence.

The defence shifts are equally noticeable; with the closure of Canadian Forces Europe as of June 94; UN peacekeeping at an all time high; national sovereignty as a major forces mission and the question of world order peacemaking participation. With new roles for the CF, the military requirement remains **the ability to fight**. Since in the present decade increases are "out", cuts are "in", Mialkowski saw a clear need for affordable technology (also dual technology), fast change (faster, smarter, cheaper) (US battle labs); innovation, and a broad, technical research base (even if modest in size). Speaking generically about technology developments, Mialkowski saw the following as being of the essence:

- rapid change
- information technology
- communication technology impacts
- surveillance technologies
- trends to larger complex systems (e.g. smart weapons)
- micro technologies
- Knowledge Based Systems
- Robotics
- biotechnology growth
- industrial production capability/ markets.

As a result of the above considerations, Knowledge-Based Systems would have to be modest, simple, practical, domain specific, useful and cheap. New system means new problems and greater complexity.

DND has no choice but to invest in Knowledge-Based Systems to deal with upward spiralling complexity in emerging technology.

Donald L. Smith, who is the Director, Command and Control Division of the Defence Research Establishment Valcartier, and an active, previous participant on this KBS panel, spoke on "Challenges and Opportunities Managing Complexity with AI Technology in Command and Control Systems". In a very comprehensive presentation he dealt with the complexity in data/information fusion and in the planning/evaluation process. He discussed in some detail the challenges and opportunities implicit in the **co-ordination** of: human factors, computer science, and operational research activities as well as the **implementation** in: verification and validation, doctrinal and organizational changes, and life cycle management.

It is perhaps worth looking a little closer at this particular argument.

Command and control is the exercise of authority and direction by a designated commander over assigned forces in the accomplishment of his and/or the forces' mission. Command and control functions are performed through an arrangement of personnel, equipment, communications, facilities and procedures, which are employed by a commander in planning, directing coordinating and controlling forces and operations in the accomplishment of the mission.[3] Command and control can be viewed as a three-stage process:[4]

a. Picture compilation - accumulation and fusion of information to provide a tactical picture;

b. Situation assessment - assessment of the situation presented by the tactical picture in relation to the mission, and

c. Response - determination of options and selection of direct action as a result of the assessment.

The complexity of command and control systems arises from factors such as:

a. their distributed nature. Command and control systems tend to be distributed in both geographical and hierarchical terms.

b. temporal constraints. Typically, command and control systems must function within hard, real-time limits.

c. uncertainty and ambiguity. The information feeding the command and control system is more or less accurate, timely, and complete.

d. multi-agent architectures. Command and control systems consist of interacting combinations of human and machine agents.

The above factors combine to make the development, implementation and life cycle management of automated command and control systems ideally suited to the application of Artificial Intelligence.

Several papers presented at this workshop addressed specific aspects of these factors and there is reason to believe that, as AI technology advances, many of the individual problems will be solved. However, the ultimate challenge remains the integration of human and machine agents into effective, intelligent command and control systems. Achieving such integration will be the subject of research for many years to come.

Conclusions

In the discussion which followed all, or most, of the points raised were dealt with, and the complementanty of the presentations was recognized. As with the previous KBS panels at earlier workshops/symposia in this field, the present special KBS panel had been specifically requested; and the evolution of the ideas and findings followed its predecessors. These points were also stressed.

By now, Knowledge-Based Systems (KBS) are probably the best known, possibly also the most frequently emphasized area of research and application within the general field of Artificial Intelligence (AI).

---

[3] ABCA Definition.

[4] Moon, J.R., Marriette, J.P. and Frampton, S.P. "Uncertainty in Command and Control Systems", Application of Artificial Intelligence to Command and Control Systems, IEE Computing Series 13, Peter Peregrinus Ltd, London, UK, 37-54.

Some years ago, Margaret Boden in a highly acclaimed work on AI postulated that, although the ability to solve complex problems in a wide variety of fields exist within AI, and in particular within ES, it was only relatively recently that the hardware costs had dropped to the point that such systems would be economically feasible.[5] This conference, and in particular the KBS panel, showed that time has in fact been compressed, and speeded up. By now these components are really becoming **embedded** in larger systems, much in the same way as electricity has become an obvious and regularized commodity of expanded usage. Nobody questions its general "validity" any more, and so it is - or should be - with Knowledge-Based Systems.

---

[5] Margaret Boden, Artificial Intelligence and Natural Man, Basic Books, N.Y. 1977.

90C210

Expert Systems in Developing Countries: Learning through IDRC's Experience

Zbigniew Mikolajuk
IDRC, P.O.Box 8500
Ottawa, Ontario, K1G 3H9.

#149711

Abstract

This paper presents some of IDRC's[†] *experience in supporting research and development of expert systems. Support for artificial intelligence applications in developing countries is a part of the program of the Information Sciences and Systems Division (ISSD). ISSD recognizes the potential role that expert system technology can play in capturing and communicating expert knowledge, including indigenous knowledge in areas such as health care, agriculture, environmental protection and decision-making support.*

Since 1986 ISSD has supported expert system research and development in Peru, Indonesia, and Philippines. Regional seminars on artificial intelligence applications were held in Africa and the Far East. Following this experience, ISSD is preparing a comprehensive plan for the support of knowledge-based technologies development. The complexity and interdisciplinary character of the program require the broadest possible collaboration and exchange of expertise among researchers from the South and the North. ISSD is particularly interested in forging links with Canadian organizations.

In this presentation we discuss issues of expertise sharing, experience in selecting proper development tools, project management and elements of the expert system development cycle. Technical problems such as expert system integration with databases and multimedia, knowledge acquisition from domain experts, and user interface are the general subjects considered in this paper. ISSD-supported projects are used to illustrate the discussed problems.

Introduction

IDRC was created by the Parliament of Canada in 1970 to stimulate and support research in developing countries. The Centre is funded entirely by Parliament to which it reports annually. Research projects supported by the Centre are identified, designed conducted, and manged by developing-country researchers in their own countries.

IDRC's Information Sciences and Systems Division (ISSD) supports research projects that enable developing countries to take advantage of applying the information technology in solving the problems of development. The challenge for ISSD is to build in developing countries the capacity to participate in research and development of the advanced technologies such as expert systems, computer-based modelling, multimedia, networking.

In this paper, we concentrate on ISSD's experience in developing and supporting projects in the area of expert systems. The availability of inexpensive expert system "shells" designed for the PC base allows for wider use of this technology and cost-effective application development. ISSD's program development policy responds to concrete needs formulated by researchers in developing countries and stimulates development of appropriate projects.

Firstly, we have learned that expert system applications need not originate only in developed countries. Secondly, the potential use of relatively simple expert systems supporting middle-level technical and professional staff may be of great importance for developing countries because of the human resources shortages at this level. Thirdly, expert system technology could play a significant role in capacity-building programs and forging collaboration of research communities of the South and the North.

[†] International Development Research Centre

TABLE I: ISSD's Expert System Projects and Activities

Year	Country	Title
1985	Argentina	IICA Expert System Consultancy
1987	Tunisia	School on Mathematics and Artificial Intelligence (ICOMIDC)
1987	Philippines	ECOKNOWMICS: A Microcomputer-based Expert System for Economic Planning and Policy-making
1988	Peru	Public Enterprise Evaluation by Expert System
1988	Senegal	ICOMIDC 2: Second School on Mathematics and Artificial Intelligence
1988	Algeria	AGROEXPERT Project Development
1988	Philippines	Workshop on Expert Systems in Agriculture
1989	Indonesia	Southeast Asian Weed Information Center (SEAWIC) - Phase II
1989	Philippines	Knowledge Acquisition Workshop IRRI
1989	Senegal	Round Table AI/RD
1990	United Kingdom	Expert Systems in Health for Developing Countries

Expert system technology was developing more slowly than it was expected by its enthusiasts in the eighties. In the mid eighties the expansion of expert system technology in developing countries was hampered by the high cost of hardware and software. At the end of the decade the inexpensive, of-the-shelf expert system shells and powerful PC technology made the cost factor less critical but insufficient awareness among decision-makers of potential benefits of expert system applications and very limited pool of people able to deliver operational systems were the unfavourable factors. On the other hand, waiting for more conducive conditions and maturity of the expert system technology to be transferred to developing countries would miss the critical point that researchers and software developers in these countries must participate in technology development as well as in application specification.

The key issue is that knowledge-based systems require knowledge acquisition from human experts. In many cases, the ways in which the knowledge is expressed and maintained are rooted in cultural environments. The understanding of the knowledge and encoding it for the processing by a computer is a task that requires the understanding of the culture and a solid background in computer technology.

Project management, access to the most advanced technologies and technical support are common problems concerning expert system developers in both the North and the South.

ISSD's Expert System Program Overview (1985-92)

In response to initiatives from developing countries ISSD's Information and Communication Technologies (in mid-eighties Information Tools and Methods) program included support for research in the area of expert system technology. Since 1985 ISSD has supported a number of seminars, workshops, surveys and software development projects concerning artificial intelligence application. Table I shows a list of these projects and activities.

Three of the projects included the development of operational expert systems for practical applications. The objectives of other activities were the capacity-building and introduction of artificial intelligence technologies.

The objective of the **"ECOKNOWMICS"** project in Philippines was to develop a prototype expert system that would provide a cost-effective solution to the problem of managing and integrating economic data and models to support planning and policy-making. This was the first project undertaken by ISSD that aimed at delivery of an expert system for a wide range of users. Unfortunately, this ambitious goal was not achieved, mainly because of underestimation of human resources required and lack of expertise in user interface development. What was obtained however, was a knowledge base representing the balance-of-payments situation. The research team could use the system in a development environment, but the end- user interface had not been developed. The researchers concentrated on a new type of methodology for representing economic models and investigation of expert system shell features. The documentation of the project was distributed to other research groups and potential users. Last July, the National Statistics Office in Philippines has expressed interest in a follow-up project.

The objective of the **"Public Enterprise Evaluation by Expert System"** project in Peru was to help support the management of public sector industry. Public enterprises constitute a very significant sector in most developing countries. Existing tools for their performance evaluation leave much to be desired. A new methodology developed by the International Centre for Public Enterprises in Developing Countries (ICPE, Ljubljana) was the base for the expert system project carried out by the Corporacion Nacional de Desarrollo (CONADE) in Peru. Two expert systems, a "Portfolio Evaluation System" and an "Enterprise Evaluation System" were built and tested on selected public enterprises. Other objectives included training, application guidelines development, testing in a real life environment, and training of enterprise managers. The last two objectives were not fulfilled, mainly because of external political reasons, frequent changes in the top management of the institution and changing view of the role of the public sector enterprises.

In the case of the **"SEAWIC"** project, the development of an expert system is a part of a larger project designed to create an information system for agricultural extension workers and farmers. The expert system component includes two modules: "Weed Identification" and "Weed Management". The first module covers 97 species of weeds in rubber plantations and contains 108 rules implemented using the CRYSTAL expert system shell. The second module is designed to give advice on preplanting, nurseries and weed treatment. Currently, it contains 111 CRYSTAL rules and new rules are being developed.

Application Identification

The selection of an expert system domain and the scope of the proposed application are critical issues in project development, implementation, and successful delivery of the software. When focusing on a proposed expert system development project, the proponents tend to ignore or underestimate the fact that in most cases the expert system is a component of a larger information system. The success of an expert system project depends on the entire application environment: data bases (quality and integrity of data), user interface features, operating system, and supporting software for graphics and image processing. On the other hand, in large information systems development, the expert system module may be treated as an additional technical "showpiece" that is given a low priority in implementation. An example of such approach was the initial status of the expert system module in the SEAWIC project. The importance of the expert system module was recognized only after creation of the herbarium data base when the usage of this data base became the design and implementation issue.

In our experience, for both types of expert system applications - standalone and embedded - the initial specifications of user requirements and functional requirements were inadequate for system verification and validation. The basic assumptions concerning the domain and scope of an application were usually not clearly identified. For example, the "Public Enterprise Evaluation ..." project's primary objective was formulated in these words: "To increase the efficiency and effectiveness of public enterprises by improving their managerial decision processes in order to ensure that they adequately fulfill their roles as instruments of national development and to prevent them from economic liabilities." The description of specific objectives and problems related to public enterprise management helped to understand the above statement but it did not provide satisfactory answers to the following questions:

- What user problem or class of problems will the expert system help to solve?
- Who will be the end users of the system?

- What is the minimum functionality required by the users?
- How will the system be upgraded and expanded in the future?
- How to measure the effects of the systems?

These issues were discussed and clarified during the project development phase involving ISSD's program officers.

Expert system user requirements and software functional specifications are often nonexistent or imprecise. In many cases, the user changes the requirements when the implementation of the system is in its final stages. The system functionality should be discussed with experts and users and fixed in the initial phase of the project. The last minute requests for changes are due to the fact that the user cannot precisely formulate his own needs, does not understand the limits of the technology and is not prepared for performing the cost analysis of required functionality in terms of human resources, proper software tools and development time.

The process of expert system application identification should involve the user, the system developer and the domain experts. The main objective of the process is to produce user requirements and functional specifications that will be usable for system verification and validation. By system **verification** we mean the process of determining whether the system provides the specified functionality. By system **validation** we mean the process of determining whether the system satisfies user requirements and performance requirements in a real environment.

What ISSD learned from the past projects can be summarized in the following points:

- Commit the resources necessary to prepare and maintain the user requirements specification.
- Determine what are the real user's problems to be solved by an expert system and state verifiable objectives.
- Determine the function and interfaces of an embedded expert system module in the context of the entire information system.
- Specify the minimum software functionality needed to satisfy user requirements and the path for upgrading and extending the system.
- Analyze similar systems and alternative technologies for development.
- Do not promise a panacea, but rather a solution of concrete problems.

Identification of an expert system application requires consultation with domain experts and computer specialists who can evaluate the feasibility of developing the proposed functionality and estimate needed resources.

Tool Selection and Knowledge Acquisition

The appearance of a wide variety of inexpensive expert system development tools greatly expanded the possibility of application development by individual researchers or small groups in developing countries.

The number of PC-based expert system shells exceeded one thousand in late eighties. Less than thirty of them dominates the market today, with NEXPERT OBJECT, LEVEL 5, KAPPA, KEE, 1-st CLASS, EXSYS, CRYSTAL, OPS-5, AION ADS, GURU, VP-EXPERT being the most popular [5].

The selection of a proper tool for application development follows the application identification process. Many publications about this subject describe all kinds of criteria and methods for tool selection. In practice, the experience of other developers and good advice in the literature do not play the expected role. In most cases, the expert system project has a unique character and the experience of others is considered not applicable. On the other hand, generic criteria for tool selection if not well understood are dificcult to use. Furthermore, the incomplete identification of expert system functionality, user requirements and scope of the project make the process of tool selection inaccurate. The problem is particularly visible in the case of the first expert system project undertaken by an organization.

Usually, in project planning the time assigned for tool analysis and learning is not sufficient. At the initial stage of the project the characteristics of required knowledge representation and expert's paradigms (the models of reasoning about a problem) are not well recognized. In effect, such factors as second-hand opinion, impressive demonstrations (easy to produce because of built-in user interface gadgets) and familiarity with a particular tool play a disproportionately large role in tool selection.

Tool selection and staff training should be done in parallel with application identification. Studying and experimenting with different tools and building throwaway prototypes do not cost very much and certainly will pay back during delivery system development. In some cases, even a very careful tool selection process may not result in an optimal choice. A change in requirements or simply learning about better tools for the project may produce the pressure to change the tool. Usually, managers of expert system projects oppose such changes, while developers support the new tool. The change of a development tool should not have dramatic or negative effects on system implementation if the system design and knowledge acquisition have been well documented.

A positive example of changing the development tool was in the ECOKNOWMICS project. The initially selected GURU package was replaced by NEXPERT OBJECT which had been found to be more suitable for modelling economic knowledge.

A development tool determines how the acquired knowledge is encoded (rules, facts, frames, decision tables, semantic networks) and which knowledge acquisition methods are applied. There are special tools for knowledge acquisition and conversion from one representation into another. In the projects mentioned above those tools were not used. Three specific areas of knowledge - economic modelling, enterprise management and botanical classification - were explored in the projects.

In all cases the knowledge acquisition process was difficult to formalize and it was not properly documented. In part, the difficulties were caused by insufficient familiarity with the expert system development methodology and development tool facilities. In one project the expert was also the knowledge engineer and principal developer; in others, larger teams of developers and external experts were employed.

Some observations and conclusions regarding knowledge acquisition can be summarized in the following points.

- Knowledge acquisition requires sufficient experience in using an expert system development tool. Project planning should include experimentation with converting the informal knowledge representation (notes from interviews with experts, manuals, drawings) into formal encoding used by a particular tool.

- Combining the learning phase with delivery system development and expanding a prototype to an operational system are common mistakes. The initial knowledge encoded for a prototype is usually not complete and properly validated. Throwaway prototyping should be included in the learning phase.

- A good relationship and communication between the knowledge engineer/developer and experts are critical for project success. The knowledge engineer must make the best effort to learn and understand the application domain. Availability of experts must be ensured and negotiated before starting a project.

- Verification and validation of the acquired knowledge must be a part of development process, not only a tool for final system testing.

Formal methods for knowledge acquisition and automatic knowledge elucidation from natural language texts are current research subjects. So far, knowledge acquisition depends on human communication skills, expertise in the application domain and experience in computer technology.

Expert System Module Integration

The final product of knowledge acquisition and knowledge base development is an expert system module that interacts with other software components of an information system - databases, spreadsheets, graphics and image files, and communication software.

The integration of an expert system into an operational environment puts additional requirements on a development team regarding expertise in computer technology.

During project planning and requirements analysis, the issue of the expertise that will be needed in areas other than knowledge engineering is often ignored or inadequately discussed. Project managers concentrate on the new technology and usually are not aware of technical details of operational environment for an expert system. Consultation with future users and personnel responsible for expert system maintenance should indicate potential problems in the area of supporting software. For example, the SEAWIC project was delayed when the development team could not solve problems concerning the integration of graphics and images with the knowledge base. The time needed for gaining the required expertise was not adequately planned.

Most of the expert system shells have built-in interfaces to standard database management software and spreadsheets. The interfaces are well tested by vendors and should not in theory cause problems. Integration problems usually appear when the expert system is integrated with user's operational database.

Some expert systems operate on external data bases created and maintained by other systems, for example a personnel database, an inventory, or census data. Usually, developers of an expert system create subsets of external databases for development testing and running demonstration versions. Under pressure to show development progress and deliver working prototypes, developers may make small changes in data bases or ignore "inconvenient" data values. Sometimes, changes are made to investigate a particular problem. These changes are often not properly documented. The net effect may be that the expert system which has been working in a development environment will not work as expected in a user environment. The identification and localization of errors can be very difficult in such a situation.

The transition from a development to a user environment and integration with operational databases should be tried as early as possible in the development process, but only for testing and evaluation of performance parameters. At this stage the system should not be used for demonstrations. The integration of a poorly tested and incomplete expert system prototype with the operational environment does not help to identify important problems but may only indicate already known difficulties.

The integration with operational databases, spreadsheets, specialized devices (e.g. scanners), and user interface procedures (e.g. data entry) requires a lot of hands-on experience. The knowledge engineer should be able to obtain support in this areas from specialists. The problem of expertise in computer technology and support for the knowledge engineer will become even more important when expert systems modules will be integrated with multimedia environments (video, voice, image processing). Voice and image processing are desired features for educational and advisory expert systems.

Program Development Strategy

Expert systems and computer-based models are the state-of-the-art components of decision-making systems and provide a platform for development of user-friendly tools for knowledge management and access. These technologies are key elements of the Information and Communication Technologies program; their application could play a significant role in fulfillment of many general and specific objectives of Agenda 21[†].

The **goal** of the ICT's Technologies for Knowledge Management and Access (TKMA) subprogram is to contribute to development of knowledge-based technologies in developing countries and to establish in the South self-sustainable research groups which will participate in both, software tools development and application development for global and local problems.

To reach this goal we should:

- identify projects which can lead to commercial products or further financial support from different sources;
- identify appropriate recipients and application domains;
- establish 2-3 research groups which will be able to continue their work without or with limited financial support from IDRC.
- deliver, as fast as, possible state-of-the-art hardware, software and training to selected research groups;
- foster working links among the expert system and modelling research teams;

The strategy for achieving the program goal can be summarized as follows:

a) Involve other IDRC divisions and IDRC's regional offices in identification of specific objectives of this program and application domains.

- *Involvement and active participation of divisions and regional offices is vital for the success of the program. Interdisciplinary character of expert system and modelling projects requires close co-operation with domain experts and potential users. Regional offices and divisions can help establish links between ISSD and institutions interested in expert system development. Their input will be critical in specifying what kind of problems the system will solve, who will be the end users, what kind of requirements regarding user interface are valid, scope and objectives of a project and validation of the system.*

b) The support will be directed at research groups which have strong computing experience and/or some experience in expert system and computer-based modelling development.

- *Expert system/model development requires concentration on the domain expertise, knowledge acquisition and software integration. It should not be distracted by the need to learn basic programming and hardware maintenance skills or database subjects. Computing problems such as interfaces with database software, graphics or video integration, user*

[†] "Agenda 21" is the final document of the United Nations Conference on Environment & Development, Rio de Janeiro, June 14, 1992.

interface development tools, and networking must be learned and resolved before starting a knowledge-based project.

c) On the recipient side a long-term commitment is required.

- *Knowledge-based system development is an interdisciplinary undertaking. Building a research team takes time. The efficiency of expert system/model development depends a great deal on communication between team members and previous experience.*

d) Foster collaboration between recipients and Canadian research institutions to accelerate the process of capacity-building.

- *In a short-term, collaboration enhances the possibility of successful implementation of an expert system because of the experience and knowledge of development tools of Canadian partners. In the long-term, collaboration can open new business perspectives for Canadian and recipient organizations (joint ventures, world-wide dissemination of information on Canadian expertise).*

e) Two types of knowledge-based system projects will be supported - capacity building and expert/knowledge-based product development.

- *The purpose of capacity building projects is to establish research groups in the environments where real needs for expert system technology has been identified; for example, at an institution which is responsible for water resources management. The purpose of product development project is to make use of established research resources to develop a self-sustainable mode of operation.*

Prerequisites for capacity-building projects:

- strong background in PC technology and database applications;
- well-defined needs for knowledge-based technology related to Agenda 21 applications;
- local expertise in application domains;
- long-term commitment to develop knowledge-based technologies and assign own resources.

Expected output of a project of this kind will be a demonstration system and establishment of a basic infrastructure for knowledge-based system development.

Prerequisites for product development projects:

- expertise in application domains;
- well-defined expert system functionality and benefits to the users;
- prior experience in expert system development;
- expertise in integration of expert system technology with other technologies (database, GIS, multimedia);
- established contacts with other research centres.

Expected output of a project of this kind will be an operational delivery system (possibly of commercial value).

Conclusions

The purpose of the presentation of IDRC efforts in supporting expert system projects in developing countries is to stimulate the North-South cooperation and exchange of ideas. The paper gives only an overview of project development problems which need further discussion and analysis.

Researchers in developing countries are interested in the application of knowledge-based technologies and ISSD will continue to support their efforts. ISSD's experience shows the need for not only the technology transfer but also closer collaboration between the South and the North in identification of application domains, feasibility studies, knowledge acquisition methods, system integration, tools development, project mangement and training.

The main elements of ISSD's program are capacity building and operational expert systems development. We expect that the program will result in establishing of self-sustainable centers of expertise which will be able to develop operational expert systems and participate in the research on knowledge-based technologies.

Bibliography

1. H. Adeli, Editor, " Knowledge Engineering", Volume I Fundamentals, McGraw-Hill Publishing Company, 1990.
2. B. Bowonder, T.Miyqake, T. Monish Singh, "Emerging Trends in Information Technology: Implications for Developing Countries", International Journal of Information Management, vol.13, no.3, June 1993.
3. M.A. Carrico, J.E. Girard, J.P. Jones, "Building Knowledge Systems: Developing & Managing Rule-Based Applications", Intertext Publications, McGraw-Hill Book Company, 1989.
4. IDRC Archives: Project files.
5. Ch. K. Mann, S.R. Ruth, "Expert Systems in Developing Countries: Practice and Promise", Westview Press, 1992.

90C 206

Learning Organizations and the Management of Knowledge

Grant Thomas
Vice President, RES Policy Research Inc.
Partner, Neurope Lab, France

RES Policy Research Inc.
6th Floor, 100 Sparks Street, Ottawa, Ontario, K1P 1B7

#149713

Abstract

Corporations around the world are re-thinking their learning and training approaches in response to the enormous pressures they face because of rapid technology, increased competition and global complexity. They are coming to understand that the knowledge, or know how, held within the organisation, individually and collectively, is perhaps their most important competitive asset. They are seeking new ways of formalizing and transacting knowledge - becoming learning organizations. New forms of user-centred, open learning environments are being trialled and new relationships with the learning industry are being sought. This paper presents a view from industry and suggests a framework to guide corporate knowledge planning. The paper is based on the author's experience in the creation of Neurope Lab, an international research consortium based in France. Neurope Lab is concerned with the application of technology to support the management of knowledge and its members include IBM, DEC, H-P, Intel, Crédit Agricole, Electricité de France, the governments of France and Switzerland and several European universities.

Keywords: Corporate learning; Just-in-time knowledge; Knowledge planning; Learning organization; Strategic planning.

Introduction

Never before has the learning industry, indeed all of those who are involved with Knowledge Based technology, been such a welcome member at the corporate table. However, is it being summoned as a partner or is it being taken to task?

Increasingly, modern organizations are realizing that their most critical competitive asset may well be the knowledge held within the company and by its employees; that corporate survival and prosperity may depend upon the organization's ability to protect, transfer and mobilize that knowledge; and that corporate futures may depend upon the ability to marshall new knowledge - *the very ability to learn.*

While organizations have always distinguished themselves by their "know how" - from construction companies to manufacturers to consultancies - new pressures make the management of that *know how* ever more important.

Global Pressures

Technology is changing at such a rapid pace that product cycles have collapsed. Global competition has forced a dramatic reduction in innovation-to-market elapsed times, eliminating the opportunity to understand and learn new technologies in conventional ways. Technology itself has become so complex that no one individual can master it in the way we used to master fields of knowledge and specific tasks. Indeed, the fusion of technologies imbedded in the average product overwhelms the ability of the modern company to manage all of the components. Juan Rada, the former head of the Lausanne-based business school, IMD, theorizes that the average car comprises 21 core technologies and that no company can itself master more than 7 technologies.

Add to this the increasing globalization of markets and we are faced with a world in which joint ventures are the norm; in which research is becoming so expensive that collaboration is the only solution; in which time pressures no longer permit the luxury of meetings and travel, thereby requiring cooperative work methods, supported by computing and communications technology; in which workers are no longer prescribed specific tasks, they now must cooperate, communicate and compute.

Knowledge Assets

The emerging collaborative paradigm requires that organizations understand fully their knowledge assets and, more importantly, understand the knowledge that they may not have but require in a given situation, in order to acquire it either through joint venture or the hiring process. Organizations must also find ways to mobilize the knowledge possessed by employees, to network this knowledge in dynamic work groups and to encourage and support an inquisitive, learning culture.

Fortune Magazine observed in its June 3, 1991 issue that "Intellectual capital is becoming a corporation's most valuable asset and can be its sharp competitive weapon". Such assets need management, formalization and protection.

Against this complexity, how are organizations reacting and what new pressures are being placed on the learning industry?

Flexible Learning

In seeking to become *learning organizations*, many corporations are re-thinking their education and training programs. The movement is strongly focused on learner-centred, open methods through which corporate goals, objectives and requirements are increasingly made visible and facilities are put in place to encourage continuous, learner-driven education.

Flexible resources are put at the employee's disposal and inquisitive networking amongst employees is encouraged. Resources may include computer based courseware, indexes to libraries, access to internal and external databases and directories of fellow employees or external consultants who may be proficient in the fields sought by a learner. These resources may be available in open learning centres - drop-in resource centres - although the trend is to the establishment of communications networks to eliminate time and space constraints. Significant research is being applied to the understanding and the establishment of open learning networks and the design of Knowledge Bases to support flexible learning. The objective is to make available *just in time knowledge*.

In one progressive case, a multinational corporation has embarked on an innovative long term program to achieve complete employee learning enablement. The program is comprehensive and will include:

The availability of a corporate long range prediction identifying desired markets and business areas, potential organization structures to meet these needs, key positions and the core competencies required of these roles.

Definition of learning programs to achieve the competencies required of future positions.

Self-assessment software to permit employees to evaluate their current competencies and their suitability for other roles.

Open learning networks enriched with resources to support the complex of learning programs.

Directories of peers, experts and tutors who are available to participate in the achievement of the learner's goals.

Organizations are realizing that, while they can no longer offer life long employment, they can offer life long learning. The concept of life long employability is replacing the single life long career. Corporations recognize that a life long learning culture will not only serve corporate knowledge needs, but will also enable the employee to manage and navigate his or her career, reducing the dependency upon institutional career planning.

Consortia Approaches

Companies are also aware that the re-engineering of corporate knowledge and learning is a very complex undertaking and that the methodologies, processes, lessons and experience are still unproven. Few organizations have the diversity of resources required to achieve progress of significance and this has resulted in the formation of research consortia to investigate and experiment in new areas of learning and knowledge based technology. Precarn Associates is a successful Canadian consortium performing joint research in intelligent systems. Neurope Lab is a European institution in which industry, academia and government research institutes are collaborating in a comprehensive agenda of research and practical experimentation in new forms of learning and knowledge management. ELEDA, a collaboration of industry and major European business schools, is looking at innovative ways of applying technology to issues of reach, productivity and quality of management education.

Knowledge Planning

While formal programs and methodologies to govern learning transformation and knowledge management are still in development, a framework to govern Strategic Knowledge Planning can be suggested. In fact, one could draw similarities from the evolution of Long Range Systems Planning.

The history of corporate information systems has seen the shift from centrally controlled mainframe computers to networked personal computers which empower the individual to meet his or her own information needs. We have seen the movement from raw data to information being respected as an asset. We have seen the data processing supervisor become the information resource executive and the Long Range Systems Plan become a key strategic document in the corporate planning process.

Similarly, within the world of corporate education programming, we are seeing centrally administered training programs give way to user-centred learning. We see knowledge emerging as an acknowledged asset, and we are seeing the appointment of intellectual property executives in some companies. Can knowledge planning be far behind?

The Strategic Knowledge Plan might have four principle components or lines of enquiry:

Corporate Knowledge Engineering

Corporate Open Learning

Knowledge Technologies

Corporate Knowledge Economy

The following is representative of the type of activity which might take place in each field.

Corporate Knowledge Engineering

The identification of current and future key knowledge requirements and critical technologies;

The key competencies required over 5, 10, 15 year horizons— of management, of professional/technical, of all staff;

Development of an inventory of knowledge assets, key corporate strengths;

Analysis of knowledge gaps and the development of a knowledge acquisition plan - internal development, acquisition, joint venture;

Development of a knowledge institutionalization strategy -development of knowledge bases, formalization of knowledge, security plan (duplicate key resources, document, capture in knowledge bases, etc.).

Corporate Open Learning

Shift to a concept of learning by design, not by decree;

Development of user-centred resources;

Development of critical path, learning road maps; Identification and encouragement of tutors, mentors and experts to participate;

Implementation of open learning facilities - networks;

Integration of external resources - business schools, universities and colleges, research community - extend corporate knowledge network.

Knowledge Technologies

Develop knowledge of critical technologies to support knowledge transfer and collaboration, corporate open knowledge architecture;

Networks, messaging, computer supported conferencing, broadband voice/video;

Multimedia, knowledge bases, hyperdocument architectures;

Supercomputing, visualization;

Human-machine interfaces, graphical languages.

Corporate Knowledge Economy

Develop concepts of valuing knowledge, depreciation, shadow balance sheet;

Develop knowledge auditing discipline;

Develop protection policies;

Develop return on investment measures;

Productivity and performance relationships;

Develop acquisition strategies - make or buy, joint ventures, mergers.

It can be argued that in the present world of seeming chaos, ambiguity and dramatic change, long range planning becomes a futile exercise. However, the described approach represents a basic framework within which a corporate knowledge and learning strategy can emerge and a pro-active plan of investment priorities can be established. The Strategic Knowledge Plan will become a key document within the corporate planning process.

Pressures on the Learning Industry

The increasing emphasis on knowledge and learning within organizations is putting pressure on the learning industry to adapt. Corporations are no longer satisfied with the residency programs offered by the business schools and, at the entry level, corporations are seeking young performers who are *knowledge adept* and not necessarily *knowledge deep* as the half life of knowledge grows ever shorter.

The business schools and vocational institutes will have to adopt similar open knowledge architectures and open learning approaches. As we move to a *just in time knowledge* environment, corporate Europe wants its educational partners to serve as nodes on their learning networks, as key incremental resources available to serve both learning and operational knowledge needs. Business will be prepared to enter into long term partnerships to personalize this technology based relationship.

Similarly, the institutes themselves must seek ways of increasing their productivity and reach while maintaining or enhancing quality. They must recognize the knowledge assets that they hold and they must implement ways to integrate them *just in time* to meet corporate and community needs.

Conclusions

Tomorrow's knowledge adept will be required to be knowledge navigators, competent in the mobilization of resources to meet changing situations and comfortable operating in dynamic collaborative networked teams. This constitutes an enormous change in traditional educational emphasis.

The transition to the knowledge age is underway in Europe. It will be realized over communications networks and will be supported by user centred open learning platforms and *just in time* knowledge architectures. It will herald a world of shifting collaborations of organizations who one day compete and are partners the next. It will reward the knowledge navigator, tomorrow's entrepreneur.

The collaboration must start now. Industry and the learning community must forge early implementations of the knowledge based cooperative model as they face their respective challenges.

904331

PANEL ON
ROBOTICS: THE PRESENT AND THE FUTURE

M. Farooq
Department of Electrical and Computer Engineering
Royal Military College of Canada
Kingston, Ontario
Canada K7K 5L0

#149714

Introduction

A special panel of experts from industry, military and universities was assembled to discuss the issues related to *Robotics: the Present and the Future*. Besides the above author, who served as the chair, the panel was composed of the following members:

1. Dr. G.M.T. D'Eleuterio, Institute of Aerospace Studies, University of Toronto
2. MGen A. Dequetville, DND (CFD)
3. Mr. James Ferguson, ISE Research Ltd
4. BGen (Ret'd) W. Niemy, Canadair
5. Dr. R.V. Patel, Dept of Electrical & Computer Engineering, Concordia University
6. Dr. S. Sachdev, SPAR Aerospace

Each member of the panel except **BGen (Ret'd) W. Niemy** made a brief presentation on the subject of robotics and its potential role in the present and the future industrial and military applications. The members of the panel also presented their views on unacceptability of robotic systems in various technological and operational areas.

Excerpts from Presentations

Dr. D'Eleuterio discussed the issue of how to make robots function in an intelligent or a smart fashion so that this technology gains acceptance in industry. He argued that the problem lies in the fact that robots are not sufficiently "intelligent"; thus, the recent spate of articles on "intelligent robotics", "smart robotics" and "autonomous robotics". A commonly accepted definition of these terms, however, is elusive. Yet, the common notion that robots must be "smarter" to provide a major impact on industry prevails. The general concept, moreover, runs the gamut of hierarchical levels from basic joint servocontrol to high-level reasoning capabilities. Let us investigate a little more closely one potential strategy that may help us towards achieving the seemingly quixotic goal of "intelligent robotics".

One area of robotic control that would seem ripe for a new approach is *manipulation control*, that is, the lower levels of robotic control dealing with trajectory control and object manipulation. Manipulation control has traditionally taken what might be termed a "parcelled" approach. That is the control algorithm is seen as one block in the overall system with specific data required from sensory devices and specific data delivered to actuators. This essentially reflects the parcelled view of intelligence where perception, reasoning and action are considered as separate functions or blocks linked together by appropriate interfaces. A competing model is that of a tightly integrated system where the functions of perception, reasoning and action are not blocks easily severable but rather are intricately interwoven. We may think of such a view as an "organic" approach to control. The term "organic control" is chosen with some care. It is intended to emphasize the "connection or coordination of parts into one whole" as well as to allude to biological structures from which this approach draws much of its inspiration.

The differences existing between organic control and conventional control are, the former is an integrated approach whereas the latter is a parcelled (or compartmental) approach, cognitive as opposed to prescribed, nonmodel-based instead of model-based.

The value of the organic approach lies in the overall goal of the development of intelligent robotics. If we are to imbue robots with "intelligence", given the various shades of meaning this word can take, we must think of a control framework that can enable, and be integrated into, the higher-level functions we associate with intelligent systems. We must therefore begin looking off the well trodden "conventional" path. We must not be as inflexible in our thinking as most of our current robots are in their operation.

MGen A. Dequetville (CFD) made an oral presentation only. He talked about the use of advance technologies in the force development, such as minimizing the risk to human life in action, etc. He expressed the same concerns as those of Colonel Mialkowski, at the previous robotics conference *(3rd Conference on Military Robotic Applications)* to emphasize the need for continued and inspired research in this area, in his keynote address, Colonel Mialkowski says:

"...It is in the development process that transforms a requirement statement into fielded equipment that robotic sub-systems and systems have been thus far introduced and I expect will continue to be evolved and integrated..."

Mr. James Ferguson talked about his familiar perspective of unmanned undersea vehicles or UUVs. As a means of understanding this perspective, it is helpful to look at the status of undersea vehicle development in relation to the development of unmanned ground, air, and space vehicles. There can be no question that the most difficult technical challenges lie in the development of ground robots, where the vehicle is continually in contact with the terrain and what is, in the other environments, a relatively straightforward navigational task becomes a continuous problem of obstacle avoidance. It is perhaps because of this, and until recently, the unquestioned availability of soldiers that development of the unmanned ground vehicle lags behind development in all other environments.

As with the air and land environments, however, there are some unique challenges and fielding UUV systems has not been a cakewalk. For starters, the operating medium is 800 times denser than air and developing energy sources which provide practical levels of endurance in manageable volumes has been an endeavour which has attracted considerable effort in the past decade - and one which is now producing measurable results. The viscous effects of seawater and dissolved trace elements absorb radio waves at ranges of less that 100 meters and acoustic transmissions at distances ranging from hundreds of meters to a few kilometres. While this encourages undersea vehicle development to be autonomous, it also limits the ability to transmit mission information in real time and because of this, a concerted effort has been made by government, industry and academia to be smarter in the way we use this transmission medium. Again, some considerable success has been achieved in the past few years.

While the focus of UUV development has been military, it is shifting. In the past decade, a number of international research programs have been undertaken to improve our understanding of the oceans and it's relation to the atmosphere. These experiments, which are described by a variety of funny acronyms such as WOCE, TOGA, LITH and GOFS, are global in nature and decadal in time. They have as their objective the development of better ocean circulation, energy and air-sea interaction models. These and numerous smaller process-oriented programs are going to provide us with the ability to do a variety of things such as forecast weather, months, rather than days in advance, better predict underwater earthquakes, and nearer to the coast forecast hypoxia and toxic blooms - both of which cause widespread disruption to the fishery. Along with this information will come a good understanding of ocean pollution, it's origins and causes.

Designing unmanned vehicles is both complex and difficult. Many components or problems such as stable physical control, motion planning, mission planning and replanning, sensing, and failure recovery must be individually solved for tractability and the diversity and dissimilarity of these problems precludes the use of simple, monolithic paradigms. Thus, if we are to obtain optimal benefit from undersea robotics in the near future, there needs be a period in the present where we can practice our integration skills and join the user in experimenting with these systems in the field. There is nothing unusual about this - the same period can be found in the development of the automobile, the aeroplane and the PC. It is also a need which seems to be well understood. In searching for a more eloquent means of expressing this need, the words of Colonel Mialkowski who made the keynote address at the last robotics conference are most appropriate. He said:

"If a picture is worth many words, then a working example is even more valuable. Commanders at all levels must have the opportunity to develop their understanding and see robotic systems in use which demonstrate that they can effectively replace a soldier within a given set of circumstances..."

In closing, we should strive to report some substantial progress at the 5th conference in 1995.

Dr. R.V. Patel discussed a number of robotics related projects and has been working in collaboration with a team from the Canadair Division of Bombardier and various faculty members in McGill and Concordia Universities. The project with Canadair Division of Bombardier, (funded under STEAR 5 Program), involves the development and implementation of state-of-the-art theoretical and practical methodologies in a number of areas in robotics - path planning, collision avoidance, redundance resolution, force and position control, object contact, hardware and software development for real-time control, etc. There are three phases to the project. Phase I was primarily a feasibility study and was completed in January 1992. Based on the work done in Phase I, contracts were awarded in 1993 for Phase II which is expected to be completed in 1995. Phase III, if awarded, will also be of two years' duration. The work involved in Phase II consists of the development of appropriate hardware and software (including simulations) required to implement the methodologies that were identified in Phase I as being the most suitable for performing the Trajectors Planning and Obstacle Avoidance (TPOA), force control and object contact. The implementation is to be done on a 7 degrees-of-freedom (d.o.f.) manipulator which is required to execute some strawman tasks. In Phase III, the technologies developed in Phase II will be extended to two 7 d.o.f. manipulators working co-operatively. By all accounts, this project has been a success in the context of technology transfer to Canadian industry in an important practical aspect of robotics.

Dr. Patel in collaboration with Professor Jorge Angeles of the Department of Mechanical Engineering at McGill University are working on the design, manufacturing, interfacing and control of kinematically isotropic redundant manipulators. This research is concerned with the design and manufacturing from scratch of robotic manipulator systems with kinematic redundancy and isotropy as the main design criteria. Isotropy offers the highest accuracy with regard to transformations between joint and Cartesian variables of motion and force. Additionally, this feature ensures the highest robustness against manufacturing and sensing errors. This, in turn, produces a larger workspace volume in which the manipulator Jacobian remains numerically well conditioned. In other words, kinematic isotropy makes a manipulator robust to singularities over a larger part of its workspace. A prototype - REDIESTRO, a REdundant, Dextrous Isotropically Enhanced, Seven Turning-Pair RObot has been developed. Algorithms to address various issues related to redundancy and isotropy, have also been developed. Design and implentation of both the hardware and software necessary to operate REDIESTRO efficiently and reliably in real-time have been carried out.

With Dr. Khorasani, Dr. Patel has been exploring the possibility of employing neural network architectures in conjunction with different control strategies for flexible-link manipulators. The team has been also working on the design of light-weight (flexible-link) manipulators. The aim of this project is to meet desirable mechanical and control design specifications so as to achieve significantly greater accuracy and more robust performance than is possible using techniques that are currently available. Such manipulators can be used in those situations where weight is a major factor.

A summary of various robotics based projects such as the famous Canadarm, the Mobile Servicing System and Inco Roof Bolter/Mining were presented by **Dr. S.S. Sachdev** of Spar Aerospace. He stated that the Canadarm was the first example of tele-manipulation in space and features sophisticated technologies such as resolved rate end point computer controlled motion, a high stiffness and low weight graphite epoxy structure, high efficiency backdriveable compound planetary gearing, samarium cobalt brushless dc motors with high resolution rate control, electro optical commutation and precision electro optical joint encoding.

The End Effector is a patented Canadian invention and features a snare wire mechanism, much like a camera's iris, for soft-capturing of free flying satellites and payloads. The Canadarm is designed to capture, manipulate and precisely stow payloads up to 65,000 lb. in mass, 15 feet diameter and 60 feet long.

During their 32 flights on Shuttle missions, the four Canadarm systems built (a fifth one has just been delivered), have performed flawlessly meeting or exceeding all performance expectations. The systems have deployed and retrieved large spacecraft such as the Long Duration Exposure Facility (LDEF), supported astronauts during spacecraft rescue missions, and knocked ice off the Orbiter's waste management system - much to the relieve of the astronauts!

The centrepiece for a significant advance in space robotic technology is the Special Purpose Dexterous Manipulator (SPDM), which is a two-armed robot with a central body. It will have the capability to be used either at the end of the Space Station Remote Manipulator System (SSRMS), or on a stand-alone basis attached to the Space Station via its body end effector. Its purpose is to carry out on-orbit maintenance of equipment on the station. It features two manipulator arms, each with seven joints, with vision systems, force moment sensing systems, and collision avoidance software built into them. A feature of the Space Station Manipulator Systems is that they are on-orbit maintainable. For example, the SSRMS is designed to permit the exchange of electric boxes and cameras robotically on orbit, as well as the exchange joints and the end effector and various other orbital replaceable units, on orbit with astronaut assistance. Each of the manipulator arms on the SPDM can also be exchanged on orbit, as can the various electronics boxes on the body of the SPDM. The systems are being designed for 30 years life with periodic maintenance. The SSRMS features fully redundant electronics and motor systems with a third hard-wired back-up electrical string for joint and end effector control.

Spar was contracted to design and develop a Remote Manipulation and Control System for the remotely controlled refurbishment of the reactors of Ontario Hydro. The centrepiece of the system was the Spar 2500 hydraulic manipulator, so called because of its 2500 lb. lift capability. This manipulator has a reach of some 40 feet, features 5 degrees of freedom and can smoothly manipulate and precisely position 30 ft. long, 2500 lb. mass tubes in a 1 g environment. This system features fully redundant hydraulic circuits for reliability and safety features that prevent uncommanded motion. Control system principles developed on the Canadarm were used in the design of this system. Although never actually used in a retubing project - Ontario Hydro in parallel developed a method of reducing radiation levels to allow manual retubing in safety - the system was fully developed, tested and delivered to Ontario Hydro.

In conclusion, Spar has been involved in the area of space robotics starting in 1974 with the application of robotics in the nuclear, terrestrial, underground and the environmental restoration areas and is planning for future applications in satellite servicing and long term mission. Robotics will continue to contribute to the advancement of science and technology in the protection of human kind from hostile environments and in extending the capabilities of humanity in these environments.

Conclusion

During the discussion period, a number of points were raised which the panel dealt with to the satisfaction of the audience. Although this was the second special panel on robotics (the first panel was held at the *Second Workshop on Military Robotics Applications* at RMC in 1989), participants raised concerns that the evolution of ideas and findings of the previous panel have not been put to practice. It was generally felt that in the areas of Advance Technologies we lack momentum and the progress is very minuscule. The participants were of the opinion that at the 5th Conference in 1995 on the subject of Advance Technologies, we should strive to report some substantial progress in utilizing these technologies both in industry and in the military.

411454

Simulation of Flexible-Link Manipulators with Inertial and Geometric Nonlinearities

Chris Damaren*
Dept. of Engineering, Royal Roads Military College
FMO Victoria, BC V0S 1B0

Inna Sharf†
Dept. of Mechanical Engineering, University of Victoria, Victoria, BC V8W 3P6

#149717

Abstract

Two important issues relevant to modelling of flexible-link robotic manipulators are addressed in this paper. First, we examine the question of which inertial nonlinearities should be included in the equations of motion for purposes of simulation. A complete model incorporating all inertial terms that couple rigid-body and elastic motions is presented along with a rational scheme for classifying them. Second, the issue of geometric nonlinearities is discussed. These are terms whose origin is the geometrically nonlinear theory of elasticity, as well as the terms arising from the interbody coupling due to the elastic deformation at the link tip. Accordingly, a general way of incorporating the well-known geometric stiffening effect is presented along with several schemes for treating the elastic kinematics at the joint interconnections. All issues are examined numerically in the context of the Space Shuttle Remote Manipulator System. The results show that all terms are required for accurate simulation of faster maneuvers. Hence, the conclusions of the paper are contrary to the previous findings on the validity of various models for dynamics simulation of flexible-body systems.

1. Introduction

Dynamics simulation of robotic manipulators with flexible links has received a great deal of attention in the last several years. This problem, although it has gained importance in industrial robotics, is particularly relevant to space applications of manipulators. Space robotic systems are made up of light-weight, large in size members and, therefore, exhibit significant structural flexibility. Moreover, ground-based experiments with these systems are difficult because of the gravitation field, and are prohibitively expensive to carry out in space. Hence, the capability to simulate the dynamics of space manipulators is essential for design, development of control strategies, as well as, real-time applications such as animation.

A number of formulations and solution algorithms have been proposed for dynamical simulation of flexible-link robots. They vary in the approach taken to develop the motion equations (as in [1], [2] and [3]), the techniques used to model the elasticity of the links [4], the assumptions made with regards to the coupling of rigid-body motion and elastic deformations [5], and, correspondingly, the complexity of the dynamics models. Despite their differences, many of the existing formulations are based on the *classical* or linear theory of elasticity.

Recently, Kane *et al.* [6] have observed that the linear theory yields erroneous motion equations for a flexible body, because they fail to account for the *geometric stiffening* effect. Indeed, this effect was identified long ago in the treatment of *planar* rotating beams [7-9]. Since then, geometric stiffening has been discussed extensively in a number of multibody problems [10,11].

In the context of flexible-link manipulator simulation, the geometric stiffening effect has been considered by Padilla & von Flotow [12], Ider & Amirouche [13] and van Woerkom [14]. In the former reference, Padilla & von Flotow justify the inclusion of the stiffening term by arguing that it is linear in the elastic displacements, and hence, is derived with a "consistent" linearization procedure. Ider & Amirouche based their formulation on a set of nonlinear motion equations. Although their procedure can be applied to systems with closed loops, they (as well as Padilla & von Fotow) present simulation results for a flexible beam and a two-link planar manipulator only. Van Woerkom discusses four techniques that can be used to account for the stiffening effect in the dynamics models and presents numerical results for a single link with tip mass. At present, there are no simulation results that demonstrate this effect on the motion of a three-dimensional, six-degree-of-freedom manipulator with elastic links.

The objectives and contributions of this paper are twofold. First, we identify the origin of and classify the different types of nonlinearities present in the dynamics equations of a chain of flexible bodies. These equations are derived by employing an *exact* (nonlinear) description of the velocity distribution in the body and *nonlinear* strain-displacement equations. Our classification is based on subdividing the nonlinear terms into three basic categories. Those in the first category represent the inertial terms that couple rigid-body and elastic motions. We refer to these nonlinearities as *inertial* since indeed they are only present when a flexible body undergoes rigid-body motion. The second category of nonlinearities comprises those that result from employment of the nonlinear theory of elasticity to describe the kinematics of elastic deformation. Following Shabana [15], we refer to these as *geometric elastic* nonlinearities since they appear in the dynamics equations in the form of nonlinear *elastic* forces. The nonlin-

---

*Assistant Professor
†Assistant Professor

ear elastic terms are derived from the strain energy of an elastic body formulated with the nonlinear strain-displacement relations and account for the geometric stiffening effect. The third group of nonlinearties, also geometric in nature, arises when considering the kinematical constraints governing neighbouring links. They can be attributed to the elastic displacement and rotation at the "tip" of one body upon interconnection with its neighbour. The rotation, in particular, can be subjected to a number of approximations, the most common of which are presented in the paper.

The second goal of the paper is to illustrate the effects of the various types of nonlinearities, for a general robotic system. To this end, we propose four approximate models, defined according to the type and order of nonlinear terms included in the equations. The simplest of these represents the linearized dynamics equations, while the complete model contains all nonlinear terms. As a numerical example, we choose a general 6-link robot modelled after the Space Shuttle Remote Manipulator System. To compare the applicability of the four aforementioned models, the results are presented for two maneuvers which yield similar trajectories, but are executed at different speeds. This selection allows us to highlight the differences between the models and, consequently, the relative importance of the nonlinear terms for various maneuver speeds.

2. Classification of Nonlinearities for a Deformable Body

Equations of Motion for a Single Body

The dynamics equations for an unconstrained deformable body $\mathcal{B}_n$ in general motion have been derived previously in a number of publications [5], [16]. Hence, we do not develop them here, but include two key relationships which are fundamental to our formulation. The first expresses the velocity distribution of $\mathcal{B}_n$ as

$$\mathrm{v}_n(\mathrm{r}_n, t) = \mathrm{v}_n(t) - (\mathrm{r}_n + \mathrm{u}_{n,e})^{\times}\omega_n(t) + \dot{\mathrm{u}}_{n,e}(\mathrm{r}_n, t) \quad (1)$$

where v_n and ω_n are the absolute velocity and angular velocity of the inboard articulation point O_n, expressed in a frame attached to $\mathcal{B}_n$ at O_n; $\mathrm{u}_{n,e}$ is the elastic displacement of the point located at r_n in the undeformed configuration. (The symbol $(\cdot)^{\times}$ denotes the 3×3 skew-symmetric matrix required to implement the vector cross product.) The absolute velocities can be collected into a single generalized velocity vector $\boldsymbol{v}_n \triangleq \mathrm{col}\{\mathrm{v}_n, \omega_n\}$ which is a 6×1 column matrix. The velocity distribution given by (1) is *exact* and it is adopted in most existing formulations.

To obtain the discrete dynamics model, the elastic deflection $\mathrm{u}_{n,e}$ is discretized according to

$$\mathrm{u}_{n,e}(\mathrm{r}_n, t) = \sum_{\alpha=1}^{s_n} \boldsymbol{\psi}_{n\alpha}(\mathrm{r}_n) q_{n\alpha}(t) \quad (2)$$

where we assume that the shape functions $\psi_{n\alpha}$ satisfy cantilevered boundary conditions at $\mathrm{r}_n = 0$. The $q_{n\alpha}(t), \alpha = 1 \cdots s_n$ are the discrete elastic degrees of freedom and s_n denotes their number. We note that contrary to what has been stated by Kane *et al.* [6], the above expansion for the displacement field does not preclude interdependence between the three components of $\mathrm{u}_{n,e}$, in particular, the "axial" and "transverse" displacements. Therefore, nor does it prohibit modelling the dynamic stiffening of the body during its motion.

With the velocity and elastic displacement distributions (1) and (2) and with the definition $\mathrm{q}_{n,e} \triangleq \mathrm{col}\{q_{n\alpha}\}$, the general equations of motion for an elastic body can be written in the form:

$$\begin{aligned} \widehat{\mathcal{M}}_{n,rr}\dot{\boldsymbol{v}}_n + \widehat{\mathcal{M}}_{n,re}\ddot{\mathrm{q}}_{n,e} &= \widehat{f}_{nT,r} + \widehat{f}_{nI,r} \\ \widehat{\mathcal{M}}^T_{n,re}\dot{\boldsymbol{v}}_n + \mathcal{M}_{n,ee}\ddot{\mathrm{q}}_{n,e} + \widehat{\mathcal{K}}_{n,ee}\mathrm{q}_{n,e} &= f_{nT,e} + \widehat{f}_{nI,e} \end{aligned} \quad (3)$$

Here, $\widehat{\mathcal{M}}_{n,rr}$, $\widehat{\mathcal{M}}_{n,re}$ and $\mathcal{M}_{n,ee}$ are the mass matrices and $\widehat{\mathcal{K}}_{n,ee}$ is the stiffness matrix. We have used the overhat notation to designate quantities which are dependent on the deformed configuration. For the "exact" model, *i.e.*, the model which contains all terms arising from (1) and (2), the mass matrix $\mathcal{M}_{n,ee}$ is the only *constant* body matrix, while in the linearized model, all body matrices are constant. The right-hand side of equation (3) contains the total generalized forces and the generalized nonlinear inertial forces, rigid and elastic. We also note that equations (3) represent a discrete counterpart of the continuous equations developed by Meirovitch [16]. The derivation of these equations via the Newton-Euler formulation is detailed in the unpublished manuscript [17].

Before proceeding to give a detailed classification of the inertial and geometric nonlinearities, it is worthwhile to mention that the present partition of the nonlinearities is not unique. For example, Kane *et al.* [6] incorporate the nonlinear kinematics of deformation through their *generalized inertia* forces. However, we believe that the classification proposed here reflects the fundamental nature of these nonlinear terms and, moreover, it follows naturally from *any* procedure for deriving the motion equations that uses (1) and (2) as the basic kinematical description.

Inertial Nonlinear Terms

The division of inertial nonlinear terms described here is somewhat arbitrary as it is not based on the order of these terms. In fact, such a classification cannot be specified uniquely, since the order of hybrid terms may be defined with respect to the order of the elastic coordinates and/or their rates and/or rigid velocities. Instead, we delimit the two groups of nonlinear terms in accordance with the two inertial terms in the motion equations to which they contribute.

The first "order" of nonlinear terms is defined so as to include the inertial forcing terms of $O(\|\dot{\mathrm{q}}_{n,e}\|\|\boldsymbol{v}_n)\|$, $O(\|\dot{\mathrm{q}}_{n,e}\|\|\mathrm{q}_{n,e}\|\|\boldsymbol{v}_n\|)$ and $O(\|\mathrm{q}_{n,e}\|\|\boldsymbol{v}_n\|^2)$. These hybrid terms contribute to the *forcing* component of the dynamics equations and are defined as follows:

$$\begin{aligned} \widehat{f}_{nT,r} &= f_{nT,r} + \delta f_{nT,r} \\ \widehat{f}_{nI,r} &= f_{nI,r} + \delta f_{nI,r} \\ \widehat{f}_{nI,e} &= f_{nI,e} + \delta f_{nI,e} \end{aligned} \quad (4)$$

where

$$f_{nT,r} = \begin{bmatrix} \int_{\mathcal{B}_n} \mathrm{f}_n \, dV_n \\ \int_{\mathcal{B}_n} \mathrm{r}_n^\times \mathrm{f}_n \, dV_n \end{bmatrix} \tag{5}$$

where

$$\begin{aligned} f_{nI,r} &= \begin{bmatrix} -m_n \omega_n^\times \mathrm{v}_n & + & \omega_n^\times \mathrm{c}_n^\times \omega_n \\ -\mathrm{c}_n^\times \omega_n^\times \mathrm{v}_n & - & \omega_n^\times \mathrm{J}_n \omega_n \end{bmatrix} \\ f_{nI,e} &= \mathrm{col}\{-\mathrm{P}_{n\alpha}^T \omega_n^\times \mathrm{v}_n - \omega_n^T \tau_{n\alpha} \omega_n\}, \quad \alpha = 1 \cdots s_n \end{aligned}$$

For completeness, we include the definition of the total generalized elastic force:

$$f_{nT,e} = \mathrm{col}\left\{ \int_{\mathcal{B}_n} \psi_{n\alpha}^T \mathrm{f}_n \, dV_n \right\}, \quad \alpha = 1 \cdots s_n \tag{6}$$

Note, that the inertial forces $f_{nI,r}$ and $f_{nI,e}$ involve only "rigid-motion" nonlinearities. The second of the modal integrals used in the definition of $f_{nI,e}$ is

$$\tau_{n\alpha} \triangleq \int_{\mathcal{B}_n} \psi_{n\alpha}^\times \mathrm{r}_n^\times \, dm_n \tag{7}$$

The hybrid (δ-) terms in (4) are determined according to:

$$\delta f_{nT,r} = \begin{bmatrix} 0 \\ \sum_{\alpha=1}^{s_n} \int_{\mathcal{B}_n} \psi_{n\alpha}^\times \mathrm{f}_n \, dV_n \, q_{n\alpha} \end{bmatrix} \tag{8}$$

$$\begin{aligned} \delta f_{nI,r} &= \begin{bmatrix} -2 \sum_{\alpha=1}^{s_n} \omega_n^\times \mathrm{P}_{n\alpha} \dot{q}_{n\alpha} \\ 2 \sum_{\alpha=1}^{s_n} \widehat{\tau}_{n\alpha}^T \omega_n \dot{q}_{n\alpha} \end{bmatrix} \\ \delta f_{nI,e} &= \mathrm{col}\left\{ 2 \sum_{\beta=1}^{s_n} v_{n,\beta\alpha}^T \omega_n \dot{q}_{n\beta} \right\}, \quad \alpha = 1 \cdots s_n \end{aligned}$$

which requires two additional modal integrals:

$$v_{n,\alpha\beta} \triangleq - \int_{\mathcal{B}_n} \psi_{n\alpha}^\times \psi_{n\beta} \, dm_n, \; \widehat{\tau}_{n\alpha} \triangleq \tau_{n\alpha} + \sum_{\beta=1}^{s_n} \Upsilon_{n,\alpha\beta} q_{n\beta} \tag{9}$$

where

$$\Upsilon_{n,\alpha\beta} \triangleq \int_{\mathcal{B}_n} \psi_{n\alpha}^\times \psi_{n\beta}^\times \, dm_n \tag{10}$$

$\widehat{\tau}_{n\alpha}$ also replaces $\tau_{n\alpha}$ in the definition of $f_{nI,e}$. In expressions (5), (6) and (8), $\mathrm{f}_n(\mathrm{r}_n, t)$ is the force distribution acting on $\mathcal{B}_n$. Through these terms, the joint control forces, $\mathrm{f}_{n,c}$, and external forces, including interbody constraint forces, enter the equation of motion [3]. We also point out that it is the $O(\|\mathrm{q}_{n,e}\| \|v_n\|^2)$ term in $\widehat{f}_{nI,e}$ which is responsible for the well-known *softening* effect in the dynamics equations.

The second group of the inertial nonlinear terms contains those terms which affect the mass matrices of the body as they represent "inertia of deformation." Thus, we define

$$\widehat{\mathcal{M}}_{n,rr} = \mathcal{M}_{n,rr} + \delta\mathcal{M}_{n,rr} \tag{11}$$

$$\widehat{\mathcal{M}}_{n,re} = \mathcal{M}_{n,re} + \delta\mathcal{M}_{n,re} \tag{12}$$

In (11), the constant rigid mass matrix is defined by

$$\mathcal{M}_{n,rr} \triangleq \begin{bmatrix} m_n 1 & -\mathrm{c}_n^\times \\ \mathrm{c}_n^\times & \mathrm{J}_n \end{bmatrix} \tag{13}$$

where m_n, c_n, and J_n are the mass, first and second moments of inertia with respect to O_n, of the *undeformed* body. The constant part of the mass matrix, which couples rigid and elastic motions is

$$\mathcal{M}_{n,re} \triangleq \begin{bmatrix} \mathrm{row}\{\mathrm{P}_{n\alpha}\} \\ \mathrm{row}\{\mathrm{H}_{n\alpha}\} \end{bmatrix}, \quad \alpha = 1 \cdots s_n \tag{14}$$

where $\mathrm{P}_{n\alpha} = \int_{\mathcal{B}_n} \psi_{n\alpha} \, dm_n$ and $\mathrm{H}_{n\alpha} = \int_{\mathcal{B}_n} \mathrm{r}_n^\times \psi_{n\alpha} \, dm_n$ are the elastic momenta coefficients.

The elastic dependence of the body mass matrix is accounted for by augmenting the first and second moments of inertia, as well as the angular momentum coefficient. This is done through $\delta\mathcal{M}_{n,rr}$ and $\delta\mathcal{M}_{n,re}$ which give rise to the hybrid terms of $O(\|\mathrm{q}_{n,e}\| \|\dot{v}_n\|)$, $O(\|\mathrm{q}_{n,e}\| \|\ddot{\mathrm{q}}_{n,e}\|)$ and $O(\|\mathrm{q}_{n,e}\|^2 \|\dot{v}_n\|)$ in the motion equations. They are defined analogously to equations (13) and (14) with:

$$\delta\mathrm{c}_n = \sum_{\alpha=1}^{s_n} \mathrm{P}_{n\alpha} q_{n\alpha}, \quad \delta\mathrm{H}_{n\alpha} = \sum_{\beta=1}^{s_n} v_{n,\alpha\beta} q_{n\beta} \tag{15}$$

$$\delta\mathrm{J}_n = \sum_{\alpha=1}^{s_n} \left[-\left(\tau_{n\alpha} + \tau_{n\alpha}^T\right) q_{n\alpha} - \sum_{\beta=1}^{s_n} \Upsilon_{n,\alpha\beta} q_{n\alpha} q_{n\beta} \right]$$

We observe that the argument of the second summation defining $\delta\mathrm{J}_n$ above is a quadratic function of elastic variables $q_{n\beta}$. Its inclusion is necessary to preserve the positive-definiteness of the augmented inertia matrix.

Geometric Nonlinearities (Nonlinear Elastic Force)

Assuming small-strain, but not necessarily small-displacement deformation, the strain energy for a linearly elastic body takes the form:

$$U_{n,e} = \tfrac{1}{2} \int_{\mathcal{B}_n} \epsilon^T \mathrm{D} \epsilon \, dV_n \tag{16}$$

where ϵ is a 6 × 1 column matrix with components of the Cauchy-Green strain tensor and D is a symmetric material properties matrix. To derive the elastic forces or the stiffness term in the motion equations, we partition the exact strain-displacement relations expressed in the undeformed coordinates into the linear and the second-order nonlinear contributions: $\epsilon = \epsilon_L + \epsilon_{NL}$. Using the above in (16) gives

$$U_{n,e} = \tfrac{1}{2} \int_{\mathcal{B}_n} \left(\epsilon_L^T \mathrm{D} \epsilon_L + 2\epsilon_L^T \mathrm{D} \epsilon_{NL} + \epsilon_{NL}^T \mathrm{D} \epsilon_{NL} \right) dV_n \tag{17}$$

which makes explicit the third- and fourth-order terms in the strain energy that arise from the nonlinear component of the strain-displacement relation. Several of the multibody dynamics formulations incorporate only the third-order contribution since it is this term that couples "axial" and "transverse" deformations, and hence accounts for the foreshortening effect.

After substituting in (17) for strains in terms of the discretized displacements, we can obtain the elastic forces by employing Castigliano's First theorem:

$$\mathbf{S}_{n,e} = \frac{\partial U_{n,e}}{\partial \mathbf{q}_{n,e}} \equiv \widehat{\mathcal{K}}_{n,ee}\mathbf{q}_{n,e} \tag{18}$$

where we have also factored the elastic force vector into a stiffness matrix and a column of elastic coordinates. In accordance with the three contributions to the strain energy, the stiffness matrix $\widehat{\mathcal{K}}_{n,ee}$ is expressed as a sum of three terms, so that:

$$\mathbf{S}_{n,e} = \left(\mathcal{K}_{n,0} + \frac{1}{2!}\mathcal{K}_{n,1} + \frac{1}{3!}\mathcal{K}_{n,2}\right)\mathbf{q}_{n,e} \tag{19}$$

In the above, the subscripts of $\mathcal{K}_n$ distinguish the order of the dependence on the elastic coordinates. This formulation of the elastic force leads naturally to a classification of the geometric *elastic* nonlinearities according to the order of the stiffness term. We also note that if one retains only the aforementioned coupling term in the strain energy and replaces the axial displacement in it by a linear function of the constant axial load, then the resulting geometric stiffness $\mathcal{K}_{n,1}$ takes a simpler, "pseudo-constant" form. In particular, it comprises terms that affect only the transverse degrees of freedom through the axial load, the latter evaluated according to the initial substitution.

Tip Deformation

The interbody geometric nonlinearities stem from the deformation of the "tip" of $\mathcal{B}_n$ with respect to the inboard articulation point. The deformed interjoint position vector is

$$\widehat{\mathbf{r}}_{n,n+1} = \mathbf{r}_{n,n+1} + \sum_{\alpha=1}^{s_n} \boldsymbol{\psi}_{n\alpha}(\mathbf{r}_{n,n+1})q_{n\alpha} \tag{20}$$

where $\mathbf{r}_{n,n+1}$ is the underformed position of the outboard articulation point O_{n+1} expressed in $\mathcal{B}_n$.

The rotation matrix from $\mathcal{B}_n$ to $\mathcal{B}_{n+1}$ can be written as

$$\mathbf{C}_{n+1,n} = \mathbf{C}_{n+1,n,r}(\theta_{n+1})\mathbf{C}_{n,e}(\mathbf{q}_{n,e}) \tag{21}$$

The matrix $\mathbf{C}_{n+1,n,r}$ includes the effects of the joint angle θ_{n+1} at O_{n+1} (for a revolute joint) and $\mathbf{C}_{n,e}$ incorporates the elastic rotation of the tip of $\mathcal{B}_n$ with respect to the inboard articulation point. This latter matrix is the solution of

$$\dot{\mathbf{C}}_{n,e} = -\mathbf{C}_{n,e}\boldsymbol{\omega}_{n,e}^{\times}\,, \quad \boldsymbol{\omega}_{n,e} = \sum_{\alpha=1}^{s_n} \boldsymbol{\theta}_{n\alpha}(\mathbf{r}_{n,n+1})\dot{q}_{n\alpha} \tag{22}$$

where $\boldsymbol{\theta}_{n\alpha} \triangleq \frac{1}{2}\nabla^{\times}\boldsymbol{\psi}_{n,\alpha}$. In the above, we have noted that $\boldsymbol{\omega}_{n,e} = \frac{1}{2}\nabla^{\times}\dot{\mathbf{u}}_{n,e}(\mathbf{r}_{n,n+1}, t)$ are the components of the angular velocity of the tip with respect to a local undeformed frame and expressed in this latter frame. This is an exact result, as noted in [18], independent of the size of the elastic displacements and rates. The price that is paid is that (22), or an equivalent parameterization, must be integrated in conjunction with the motion equations.

For small elastic displacements and rates, one can further write

$$\mathbf{C}_{n,e} = \mathbf{1} - \boldsymbol{\theta}_{n,e}^{\times}\,, \quad \boldsymbol{\theta}_{n,e}(t) \triangleq \sum_{\alpha=1}^{s_n} \boldsymbol{\theta}_{n\alpha}(\mathbf{r}_{n,n+1})q_{n\alpha}(t) \tag{23}$$

which is directly obtainable from the elastic displacements $\mathbf{q}_{n,e}$. Implicit in this first-order approximation is the assumption that $\boldsymbol{\omega}_{n,e} = \dot{\boldsymbol{\theta}}_{n,e}$, *i.e.*, a rotation can be treated as a vector. Other authors have assumed that $\boldsymbol{\theta}_{n,e}$, as defined by (23), can be assigned to a particular sequence of Euler angles. For example, Kane *et al.* [6] assume a 1-2-3 Euler sequence. For noninfinitesmal displacements, however, $\mathbf{C}_{n,e}$ will depend on the order of rotation selected, a choice which may not be obvious for three-dimensional modelling and moreover, is arbitrary. In this case, identification of $\boldsymbol{\theta}_{n,e}$ as an Euler angle set precludes the equation of $\dot{\boldsymbol{\theta}}_{n,e}$ with $\boldsymbol{\omega}_{n,e}$. Hence, (22) would be replaced with $\boldsymbol{\omega}_{n,e} = \mathbf{C}_{n,e}^{T}\mathbf{E}_n\dot{\boldsymbol{\theta}}_{n,e}$ where $\mathbf{E}_n$ is the transformation from Euler rates to angular velocity (expressed in the *deformed* tip frame). For planar motion, (22) and the Euler angle interpretation are equivalent.

Kinematical Constraints for a Chain of Bodies

The generalized velocities of a chain of interconnected bodies can be written recursively [3] as

$$\boldsymbol{v}_{n+1} = \mathcal{T}_{n+1,n}\boldsymbol{v}_n + \mathcal{S}_{n+1,n}\dot{\mathbf{q}}_{n,e} + \mathcal{P}_{n+1}\dot{\theta}_{n+1} \tag{24}$$

where

$$\begin{aligned} \mathcal{T}_{n+1,n} &\triangleq \begin{bmatrix} \mathbf{C}_{n+1,n} & -\mathbf{C}_{n+1,n}\widehat{\mathbf{r}}_{n,n+1}^{\times} \\ \mathbf{O} & \mathbf{C}_{n+1,n} \end{bmatrix} \\ \mathcal{S}_{n+1,n} &\triangleq \begin{bmatrix} \mathbf{C}_{n+1,n}[\mathrm{row}\{\boldsymbol{\psi}_{n\alpha}(\mathbf{r}_{n,n+1})\}] \\ \mathbf{C}_{n+1,n}[\mathrm{row}\{\boldsymbol{\theta}_{n,\alpha}(\mathbf{r}_{n,n+1})\}] \end{bmatrix} \end{aligned} \tag{25}$$

and $\mathcal{P}_{n+1}$ is a projection matrix which aids in expressing the generalized velocity induced by the joint motion. Although (24) has been written in a form restricted to single DOF revolute joints for notational clarity, multi-DOF joints encompassing translational and/or rotational behaviour are easily handled [3].

It is through the generalized transformation matrices of (25) that the different approximations for tip deformation give rise to interbody nonlinearities. In particular, four possible modelling approaches present themselves:

(a) Neglect the kinematical nonlinearities which accrue from elastic deformation. This is equivalent to setting $\mathbf{C}_{n+1,n} \equiv \mathbf{C}_{n+1,n,r}(\theta_{n+1})$ and $\widehat{\mathbf{r}}_{n,n+1}^{\times} \equiv \mathbf{r}_{n,n+1}^{\times}$ in Eq. (25).

(b) Model the elastic effects to first order by using the approximation (23) and the exact expression (20) for the interjoint position.

(c) Take the elastic rotations $\boldsymbol{\theta}_{n,e}$ to represent an Euler angle sequence. In this case, the bottom partition of $\mathcal{S}_{n+1,n}$ must be modified to read $\mathbf{C}_{n+1,n,r}\mathbf{E}_n[\mathrm{row}\{\boldsymbol{\theta}_{n,\alpha}(\mathbf{r}_{n,n+1})\}]$ and $\mathbf{C}_{n,e}(\boldsymbol{\theta}_{n,e})$ must be constructed appropriately.

(d) Use the exact modelling implied by (20) and integrate (22) for the true orientation of the tip, $\mathbf{C}_{n,e}$.

The recursive relationship governing the generalized acceleration can be acquired by differentiating (24). The particular form of the matrices $\dot{\mathcal{T}}_{n+1,n}$ and $\dot{\mathcal{S}}_{n+1,n}$ depends on the kinematical modelling adopted.

3. Approximate Dynamics Models

With the proposed classification of the nonlinear terms, we now define four dynamics models, each distinguished by the "order" of the nonlinear terms retained and the approximation used for the interbody constraints.

Our first and simplest model is obtained by a ruthless linearization of the dynamics equations. It contains none of the inertial, nor elastic, nor interbody nonlinearities and is therefore defined using the constant body matrices and the "unhatted" forcing terms in (3). Accordingly, in the kinematics constraints employed with these equations we neglect the elastic deformation at the tip. Following Reference [12], we refer to this model as *ruthlessly linearized* and denote it by RL.

The second model worthy of separate designation is the one which we call *inconsistent* (I). It incorporates all of the inertial nonlinearities, but neglects the geometric elastic nonlinearities. This model is very similar to the *inconsistently linearized* model of [12], which in turn forms the basis of the majority of multibody dynamics codes. The main difference between the two is that our model includes second-order mass matrix corrections and therefore is not linear. The inconsistent approximation of the dynamics equations is usually combined with the first-order interbody kinematics as described in (b) of the previous subsection. However, in our numerical example we also present results for this model in combination with the other two descriptions of the interbody kinematics—the 3-2-1 and 1-2-3 Euler sequences, and the exact model as specified in (c) and (d). These will be denoted by I(1-2-3), I(3-2-1) and I(E).

The third model considered here is the *consistently linearized* approximation. It includes the inertial hybrid forcing terms (without mass matrix corrections) and the geometric stiffness matrix derived under the constant axial load assumption. The tip deformation for the interbody kinematics is modelled to first-order and the resulting model is denoted by CL.

Our last model is based on the *exact* dynamics equations for an elastic body. These include all of the inertial and geometric elastic nonlinearities and therefore contain terms up to third-order in elastic coordinates. The nominal description of the interbody kinematics employed with the exact motion equations will be, naturally, the exact geometrical model of (d), although we will also investigate their performance in conjunction with the linear and Euler sequence approximations. The resulting models will be referred to as EEE, EE(1), EE(3-2-1) and EE(1-2-3).

Finally, we note that a solution of the simulation dynamics problem requires determination of $\{\ddot{\theta}_n, \ddot{q}_{n,e}\}$ at each time step given the control torques at each joint. In general, this can be accomplished via two families of procedures—global techniques and recursive ones. The results given in this paper have been computed using the recursive dynamics procedure documented in [19]. In this procedure, the net (constraint plus control) interbody force is eliminated by assuming it to be an affine function of the generalized accelerations. It should be observed that the nonlinear terms, as presented in §2, do not alter the basic form of the key relationships (3) and (24) which form the basis of most recursive algorithms.

4. Numerical Results

The goal of the present section is to evaluate the performance of the models introduced previously on a 6-DOF manipulator, modelled after the Space Shuttle Remote Manipulator Arm. Included in the model is the Space Shuttle—body $\mathcal{B}_0$ of the chain—although, for the numerical results presented here, its motion is constrained. Furthermore, the arm is assumed to carry a payload, modelled by a cylindrical drum to represent a spin-stabilized satellite.

The properties of the members of the system are summarized in Table I and the architecture of the Arm is shown in Figure 1. We point out that the Space Shuttle and the Payload are approximated by rigid bodies. The only flexible bodies are links 2 and 3 which are the lower and upper arm booms. They are modelled using six natural constrained modes for each boom: two bending modes in each of the in-plane and out-of-plane directions, one stretch mode, and one torsional mode. The fundamental frequencies of the two booms are 37.2 rad/s and 34.5 rad/s, respectively.

Table I: Properties of the SRMS

Property	ℓ (m)	Mass (kg)	J_n^r (kg·m²)	J_n^p (kg·m²)	J_n^y (kg·m²)
Shuttle	–	93, 270	1.17×10^6	9.1×10^6	9.5×10^6
Link 1	0.9	95.0	0.2	25.75	25.75
Link 2	6.4	138.0	0.4	1884.36	1884.36
Link 3	7.0	85.0	0.4	1388.53	1388.53
Link 4	0.5	8	0.2	0.76	0.76
Link 5	0.8	44	0.2	9.49	9.49
Link 6	0.6	41	0.2	5.02	5.02
Payload	–	15, 000	30, 000	515, 000	515, 000

Stiffness	EI (N·m²)	GJ (N·m²)	EA (N)
Link 2	4.046×10^6	2.040×10^6	2.790×10^9
Link 3	2.812×10^6	1.417×10^6	1.194×10^9

The comparison of the different models will be done for maneuvers generated with the control torques calculated using the inverse dynamics equations for the rigid-body model of the Arm. The prescribed joint angles are parametrized by:

$$\theta_n(t) = 0.4 \times \left(\frac{t}{T} - \frac{1}{2\pi}\sin\frac{2\pi t}{T}\right), \quad n = 1, \ldots, 6 \quad (26)$$

which represents a generic "pick-and-place" maneuver for the duration time T. We consider two maneuvers—fast (F) and slow (S)—obtained by setting $T_F = 20$ sec and $T_3 = 60$ sec, respectively. We note that the maximum joint rate reached during the fast maneuver is 0.04 rad/sec which is three orders of magnitude lower than the fundamental frequencies of the booms. The maximum end-effector speed is 1.3 m/s and it is approximately an order of magnitude higher than the present speed attained during operation of the Shuttle Arm.

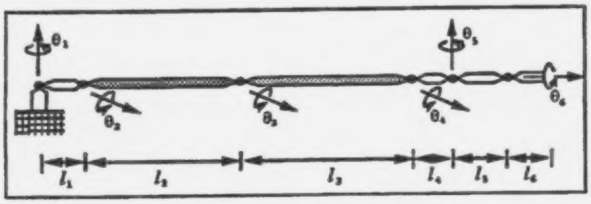


Figure 1: Architecture of the SRMS

The validity of the selected models can be assessed by measuring the error in the energy balance for the system. We use the *root mean square* value of the percentage energy balance error, the latter calculated at time t_i as:

$$e_E(t_i) = \frac{E(t_i) - W(t_i)}{|E_{peak}|} \times 100\% \tag{27}$$

In the above, E is the total energy of the system, E_{peak} denotes the maximum total energy reached during the maneuver and W is the work done on the system by the control forces. The RMS value of the percentage energy drift is then determined according to:

$$e_{E,\mathrm{RMS}} = \sqrt{\frac{\sum_{i=1}^{1000} e_E(t_i)^2}{1000}} \tag{28}$$

The values of $e_{E,\mathrm{RMS}}$ obtained with the four dynamics models for the two maneuvers are listed in Table II.

Table II: RMS Energy Drift for SRMS Example

Run	Model RL	Model CL	Model I(E)	Model EEE
F	∞	∞	4.3×10^{-5}	2.6×10^{-4}
S	12.2	1.6×10^{-4}	2.1×10^{-7}	6.4×10^{-7}

The simulations using models RL and CL for Maneuver F failed to finish owing to floating point overflow conditions. Hence, the value of $e_{E,\mathrm{RMS}}$ has been tabulated as ∞ in Table II. From the table we conclude that there is little to distinguish the inconsistent and exact models for both maneuvers on the basis of energy drift. For the slow maneuver—the only case where RL and CL models produce results for the complete trajectory—the high energy drift for the former indicates the doubtfulness of the solution. Although the drift for the CL simulation of this maneuver is not as good as that for the EEE (or I(E)) model, an inspection of the corresponding joint angles will reveal little difference between them.

The joint angle histories for maneuver F are given in Figure 2 for the EE model. Only two joint angles are given on account of space limitations. The four kinematical schemes of §2 yield similar behaviour but there are discernible differences in the last 5 seconds of the simulation. More surprising are the results for the I model given in Figure 3. The influence of the elastic kinematics breaks down into two distinct groupings. The exact and 3-2-1 Euler sequence yield similar curves while the linear approximation and 1-2-3 sequence provide completely different behaviour. These curves would suggest that a 3-2-1 approximation — the torsional rotation is performed "last" — is superior to 1-2-3 sequence if integration for the elastic tip attitude is to be avoided. They also indicate that it is the omission of the stiffening terms in the inconsistent model which amplifies the differences in how the tip rotation is handled. Indeed, this is in accordance with our expectations, since the geometric stiffening terms effectively increase the bending stiffness of the links and consequently, the stiffened (EE) model predicts smaller elastic deformations than the inconsistent one. These evidently make a substantial difference, even though the joint rates are much lower than 10% of the fundamental frequencies of the elastic booms. Moreover, one could not have predicted such a discrepancy from the corresponding energy drift values.

The joint angles for maneuver S are given in Figure 4. Models I, EE, and CL give almost identical results, independent of the method used to describe the rotational kinematics of the tip of links 2 and 3. The RL model generates radically different results and examination of the joint rates revealed an instability, which is also reflected in the relatively large value of $e_{E,\mathrm{RMS}}$ in Table II. This particular run demonstrates vividly the importance of verifying the simulation results with a physically meaningful measure, such as the energy drift for the system. However, as evidenced by maneuver F for models I(E) and EEE, the energy check by itself is not sufficient to validate the solution.

5. Concluding Remarks

In this work, several aspects of the dynamics modelling and simulation of flexible manipulators have been discussed. We presented and classified the inertial and geometric nonlinearities that arise in the motion and constraint equations for multibody systems. Subsequently, four models were proposed ranging from the simplest approximation in which none of the nonlinear terms are retained to the exact model where all terms originating from the fundamental nonlinear kinematics relations are included. It is clear from the results that both the inertial and geometric nonlinearities can have a profound effect on the simulation. It is hoped that the paper serves to partially answer the question "which terms in the motion equations are important?"

Although the specifics of the results will change when different manipulator systems are considered, we can make a few generalizations based upon our findings for a general 6-DOF manipulator. For sufficiently fast maneuvers of a realistic duration, the *ruthlessly* linearized approximation to the manipulator dynamics is wholly inadequate. In many cases, the deficiencies can be compensated for by including the nonlinear inertial forces which gives the *inconsistently* linearized model. The *consistent* model, which incorporates most

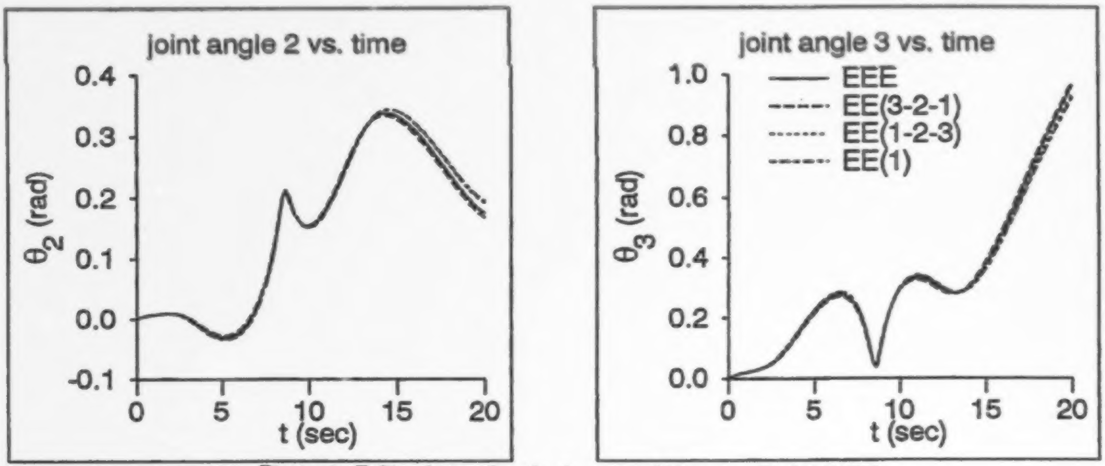


Figure 2: F Simulation Results for SRMS (*Maneuver* F, *Model* EE)

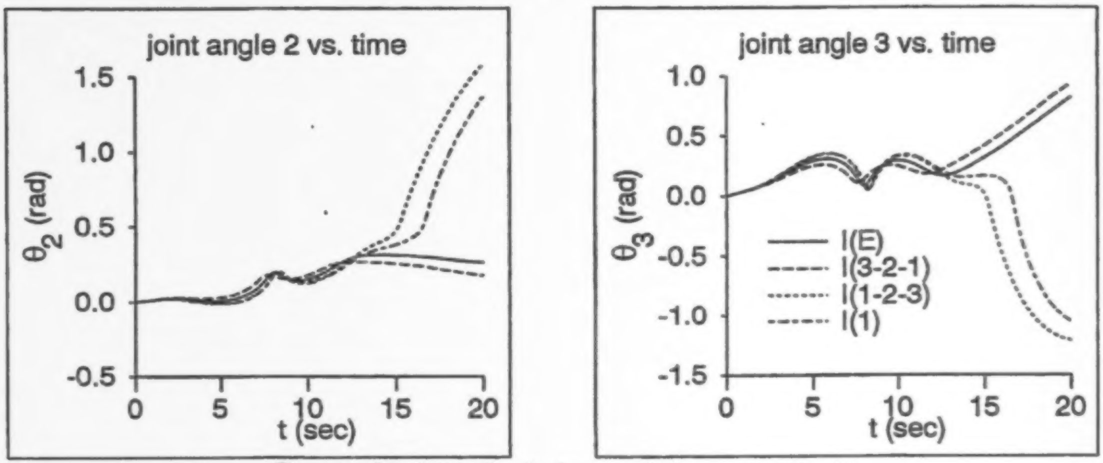


Figure 3: Simulation Results for SRMS (*Maneuver* F, *Model* I)

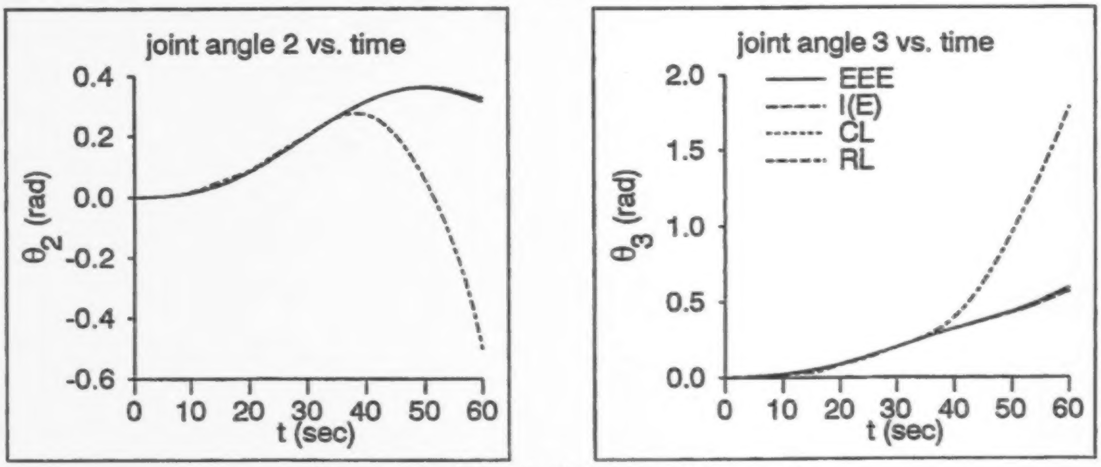


Figure 4: Simulation Results for SRMS (*Maneuver* S)

of the inertial nonlinearities and the "first-order" stiffening term, performs as poorly as the ruthless model. These results are radically different from what one observes for a single beam or a planar two-link manipulator and hence, they are contrary to the findings of Padilla & von Flotow [12]. We suggest that the ruthless model, strongly advocated in the aforementioned reference, fails to provide a satisfactory approximation for the dynamics simulation of a general manipulator system. On the other hand, the inconsistent model presents an alternative set of equations which yields reliable simulations for a wide range of maneuvers and joint speeds. In many cases, it yielded reasonable results when both the ruthless and consistent models failed. We also discovered that the additional inclusion in the I model of the elastic coordinate dependence in the mass matrix insures its positive-definiteness and significantly reduces the magnitude of the energy drift.

It was shown in the paper that the geometric stiffening terms may have a considerable effect on the solution even in cases where the fundamental frequencies of the elastic links are well below the 10% value of the angular speeds. We have also demonstrated the necessity of retaining the third-order stiffening terms in the dynamics equations and not only the geometric stiffness that couples axial to transverse deformation, as done in several multibody formulations.

In terms of modelling tip deformation for the interbody kinematics, our numerical results proved that the common assumption of infinitesmal elastic rotations is not appropriate for simulating faster maneuvers. In fact, a linear approximation was often unreliable when compared with an exact solution. However, treating the rotations as the elements of an Euler sequence yielded reasonable results. In particular, we suggest that a 3-2-1 sequence for beams may be superior to a 1-2-3 choice when torsional degrees of freedom are present. The consequences of this are important since a "good" Euler sequence will allow one to avoid the exact treatment which requires numerical integration for the "tip" attitude. We also observed that inclusion of stiffening in the dynamics model mitigates the differences between various kinematical schemes.

References

[1] Serna, M.A. & Bayo, E. (1987). "A Modified Lagrangian Formulation for Elastic Robots," *Proc. 2nd International Symposium on Robotics and Manufacturing: Research, Education, and Applications*, New Mexico, pp. 513-520.

[2] Wehage, R.A. & Shabana, A.A. (1989). "Application of Generalized Newton-Euler Equations and Recursive Projection Methods to Dynamics of Deformable Multibody Systems," *Proc. 1989 ASME Design Technical Conferences—15th Design Automation Conference*, Montreal, September, 19-3, pp. 17-25.

[3] Sincarsin, G.B. & Hughes, P.C. (1990). "Dynamics of Elastic Multibody Chains: Part A—Body Motion Equations; Part B—Global Dynamics," *Dynamics and Stability of Systems*, **4**, 3 & 4, pp. 209-243.

[4] Naganathan, G. & Soni, A.H. (1988). "Nonlinear Modeling of Kinematic and Flexibility Effects in Manipulator Design," *J. of Mechanisms, Transmissions, and Automation in Design*, **110**, September, pp. 243-254.

[5] Nagarajan, S. & Turcic, D.A. (1990). "Lagrangian Formulation of the Equations of Motion for Elastic Mechanisms With Mutual Dependence Between Rigid Body and Elastic Motions. Part II: System Equations," *J. of Dynamic Systems, Measurement and Control*, **112**, June, pp. 215-224.

[6] Kane, T.R., Ryan, R.R. & Banerjee, A.K. (1987). "Dynamics of a Cantilever Beam Attached to a Moving Base," *J. Guidance, Control, and Dynamics*, **10**, 2, pp. 139-151.

[7] Likins, P.W., Barbera, F.J., & Baddeley, V. (1973). "Mathematical Modeling of Spinning Elastic Bodies for Modal Analysis," *AIAA Journal*, **11**, pp. 1251-1258.

[8] Vigneron, F.R. (1975). "Comment on 'Mathematical Modeling of Spinning Elastic Bodies for Modal Analysis'," *AIAA Journal*, **13**, pp. 126-127.

[9] Kaza, K.R. & Kvaternik, R.G. (1977). "Nonlinear Flap-Lag-Axial Equations of a Rotating Beam," *Acta Astronautica*, **15**, 6, pp. 1349-1360.

[10] Banerjee, A.K. & Lemak, J.M. (1991). "Multi-Flexible Body Dynamics Capturing Motion-Induced Stiffness," *Trans. ASME J. Applied Mechanics*, **58**, pp. 766-775.

[11] Walrapp, O. (1991). "Linearized Flexible Multibody Dynamics Including Geometric Stiffening Effects," *Mechanics of Structures and Machines*, **19**, 3, pp. 385-409.

[12] Padilla, C. E. & von Flotow, A. H. (1992). "Nonlinear Strain-Displacement Relations and Flexible Multibody Dynamics," *J. Guidance, Control, and Dynamics*, **15**, 1, pp. 128-136.

[13] Ider, S.K. & Amirouche, F.M.L. (1989). "Nonlinear Modeling of Flexible Multibody Systems Dynamics Subjected to Variable Constraints," *Trans. ASME J. Applied Mechanics*, **56**, pp. 444–450.

[14] Van Woerkom, P.Th.L.M. (1991). "Modified Dynamics Modelling for Maneuvering Flexible Space Manipulators," *42nd Congress of the International Astronautical Federation*, IAF-91-356, October, Montreal.

[15] Shabana, A.A. (1991). "Constrained Motion of Deformable Bodies," *International Journal for Numerical Methods in Engineering*, **32**, pp. 1813-1831.

[16] Meirovitch, L. (1991). "Hybrid State Equations of Motion for Flexible Bodies in Terms of Quasi-Coordinates," *J. Guidance, Control, and Dynamics*, **14**, 5, pp. 1008-1013.

[17] D'Eleuterio, G. M. T. (1987). "Notes on Nonlinearities in Flexible-Body Motion Equations."

[18] Spencer, A. J. M. (1980). *Continuum Mechanics*, Longman, London.

[19] Damaren, C. J. & D'Eleuterio, G. M. T. (1989). "On the Relationship Between Discrete- Time Optimal Control and Elastic Multibody Dynamics," in *Contemporary Mathematics: Dynamics and Control of Multibody Systems*, **97**, American Mathematical Society, Providence, RI.

Y05C10

A Design Methodology for the Development of Experimental Robotic Systems

S.J. Miller[1] & J.-C. Piedboeuf [2]
Department of Mechanical Engineering, Royal Military College of Canada
Kingston, Canada K7K 5LO

#149719

Abstract

From a research perspective, it is almost impossible to purchase a robot that will satisfy all the constraints of an experiment. Therefore at the Royal Military College, we have designed our own robots for various experimental investigations. From this experience, we have determined four major areas of importance in the design of a robotic system. In chronological order these areas are i)robot mechanics, ii)power supply, iii)sensors and iv)computer data acquisition and control.

This paper will detail how we have designed our robotic systems using this sequence of analysis. Furthermore, we will provide justification for the selection for all components(motors, power supplies etc.) from a range of different types present in todays market.

Introduction

One of the most discouraging aspects of robotic research is the extreme difficulty in attaining an operable experimental laboratory that meets the needs of the investigation. Most often, researchers are left to purchase an entire robotic system and forced to cope with the limitations inherent to it. Such systems tend to have a "black-box" configuration and are not well suited to changes or enhancements from one research application to another. Furthermore, any mechanical/electrical problem which may occur will be difficult to diagnose and leads to lengthy down times.

A more desirable alternative would be to have the researcher take an active role in designing the robotic system. He/she could then ensure that the system will meet the required experimental constraints. Such an approach would allow regulation over the dynamic response, sensor measurement, data acquisition and changes in controller implementation. Furthermore, the system may be designed to be modular in assembly and allow ease of maintenance. A robot designed under these principles would be easy to adapt to different experiments and allow easy diagnosis if failure should occur.

The latter approach to experimentation has been our objective in the design of three robotic systems. This paper will detail the chronological methodology that was common to all the projects. Although the content of this paper is by no means unprecedented, its purpose is to provide a broad discussion on all aspects which should be considered in the design of an experimental robot. Lastly, we do attempt to keep the discussion general so that it may be applied to most robotic systems irrespective of its actual purpose.

Robot Mechanics

The most important stage to the design of a robotic system begins with the analysis of the robot mechanics. By robot mechanics, we mean the large area which covers both the dynamics and strength(stiffness) calculations of the system. This first stage of design is crucial for it provides valuable information on the size and type of components that will be necessary.

Dynamic Analysis

At this point, a designer must have a clear conception of the constraints for the robot. For the dynamical analysis we need to have an estimates of:

---

[1] Phone (613) 541-6685

[2] Phone (613) 541-6309

- i) the number of joints (rotary or prismatic) needed and the kinematic arrangement,
- ii) the size of the workspace to estimate the length of the link(s),
- iii) An estimate of the mass and inertia of the link(s), actuator(s) and the end effector load, and
- iv) the maximum range of motion and the minimum time required to complete it.

Note some of these constraints are not entirely obvious or easily known. However, these are just estimates. This design analysis is an iterative approach and we may have to complete several loops to achieve the desired robot. Therefore keep the estimates simple. Estimates of linkages as hollow cylinders and actuators as solid cylinders are satisfactory. Make use of manufacturers catalogues which may specify some of these characteristics. Lastly, we wish to stress that constraint (iv) will determine the absolute dynamic response of the robot. Thus the designer should choose the most demanding trajectories for the robot to follow.

From constraints (i) and (ii), one can calculate the inverse kinematics[1] of the robot with respect to some inertial frame. Using constraint (iv) and the inverse kinematics of the robot, the designer may then calculate the initial and final positions of each actuator. Then, a trajectory for each joint may be calculated for example:

$$q_i(t) = q_0 + 3(q_f - q_0)t^2 - 2(q_f - q_0)t^3 \tag{1}$$

where q_0 and q_f are the initial and final positions respectively of joint i. By differentiating (1) with respect to time, we then obtain the velocity and acceleration information for each joint. Lastly using the estimation (iii), we calculate the required generalized force τ of each joint through the dynamic equation[1]:

$$\tau = D(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q) \tag{2}$$

The solution of (2) will provide the designer with a profile of the force required by each joint to complete the trajectory specified in (1). The peak force from this profile and the maximum velocity defined by the trajectory may be used to select the actuator.

Note that the solution of (2) may be very simple for a one or two degree of freedom system but will quickly increase in complexity as more degrees of freedom are added. To accommodate for this complexity, it would be best to generate a general simulation program which would solve the dynamic equation. Ideally, the program should allow the kinematic, mass and trajectory parameters to be read into the simulation from a file. We have developed such a simulation program written in FORTRAN here at the RMC.

It is important to remember though that the solution of the dynamic equation only provides an estimation of the torque required to move the robot. In all of this dynamic analysis, there was no consideration given to any frictional forces or transmission efficiencies that will by inherent in the system. It is typical that the break away friction on a DC-servo motor is 4-6% the full rated torque[2] and that the efficiency of a harmonic drive is roughly 85%[3]. Therefore, this amount of torque should be added to calculated torque. If the designer were to choose a motor with a rated torque that was close to the calculated torque, all subsequent design would be limited by the masses and inertias estimated in (iii). Therefore a designer should select an actuator that is generally 20% greater than the calculated torque. This will allow him/her some leeway in subsequent design to change the masses and inertias if necessary. However, one should consult with motor companies to determine whether an existing motor is available that would be capable of producing the required torque and possesses mass and inertia properties comparable to the simulated actuator. At this point the designer may be forced to reduce either (or both) the mass of the robot or the size of the workspace and reiterate the dynamic analysis until a suitable actuator can be found.

Selection of Actuators

The three main sources of power for an actuator are hydraulic, pneumatic or electrical. The main difficulty in using either a hydraulic or pneumatic actuator for an experimental robot is that there is no general model of either actuator that is universally accepted whereas there are very simple linear models of for an electrical actuator. However, of all the three sources of power, the hydraulic actuator will generate the largest torque and force. Therefore, if a robot requires very large forces and torques in its dynamics, then a designer may be forced to use a hydraulic actuator.

Most robotic actuators are electrically powered from either an AC or DC source. From our experience, we find the use of DC motors best suits our applications. The main reason being that a DC motor may have linear control of either the output speed or torque depending on the type of amplifier used. If a current amplifier is used, the output torque is directly related to the current[4]:

$$\tau = K_t I \tag{3}$$

where K_t is the torque constant for the motor. If a voltage amplifier is used, the output speed is directly related to the applied voltage[4]:

$$\omega = K_v E_a \tag{4}$$

where K_v is the speed constant for the motor. Note that in both equations (3) and (4) it has been assumed that the flux per pole of the motor remains constant which is a valid assumption.

Due to the relatively small amount of torque produced by an electric motor, it will be necessary to amplify the torque through some form of high ratio reducer. With the proper selection, a reducer will allow a smaller motor to develop the required torque output, thus reducing the mass of the joint. Moreover, the reducer will help reduce the reaction torque applied to the motor through the relationship[5]:

$$\tau_{MOTOR} = \frac{\tau_{DIST}}{N} \tag{5}$$

where N is the reduction ratio of the reducer.

Note that the designer does have the option to use a direct drive motors(ie. no reducer) for the robot design. However, the motors will tend to be relatively larger and heavier. Such designs would be useful only in horizontal planes where gravity will not effect their motion.

Selection of Reducers

There are two types of reducers which are commercially available which are suitable for robotic joints. They are i)planetary gear trains and ii)harmonic drives. Both are relatively compact in size (low mass), efficient in torque transmission (>85%) and both have the input and output shafts concentric to each other. However, there will always be a relatively large backlash inherent in a planetary gear train. The backlash is a nonlinearity in a system which can lead to the development of a limit cycle[6] which will limit the gains in the controller. On the otherhand, a harmonic drive has a backlash that is very small (typically 1 arc minute[3]) which creates a more stable system. However, a harmonic drive will have a lower torsional stiffness due to the use of a flexible spline in the drive train. Still this reduction in torsional rigidity can be advantageous when dealing with joint torque measurement.

The measurement of torque output from any robotic joint implies that some torsional compliance must be present. Since such a compliance already exists in a harmonic drive, several inventors have developed a method to measure the output torque through the strain measurement of the flex spline[7]. With this method, the output torque of the joint varies linearly with the output voltage of the strain gauge amplifier.

When selecting an harmonic drive, the designer must make sure that it will be capable of transmitting the maximum torque and speed that were calculated in the dynamic analysis.

Strength and Stiffness Analysis

Once the motors and reducers are selected, the designer may then begin calculating the strength and stiffness of the linkages and joint housings. However, we are assuming that some general sketch or drawing of the desired joint and linkages is completed. The sketches need not contain much detail. All that is crucial for the strength design is general size and assembly. We are overtly ignoring the selection of the joint assembly because it is heavily dependant on the configuration of the robot.

Link Design

The four main concerns for the design of a robot link are deflection, strength, natural frequency and material. In this section we will demonstrate how to calculate (or choose) these characteristics of a link to the designers specifications.

Deflection Analysis

We consider the case of a single link with an endpoint mass which experiences a constant angular acceleration $\ddot{\theta}$ centred at the base of the link. The acceleration may be easily estimated from the dynamic analysis. The bending moment caused by the angular acceleration is[8]:

$$M_x = \rho\ddot{\theta}\left(\frac{L}{2}(L-x)^2 - \frac{(L-x)^3}{6}\right) + m_e\ddot{\theta}(L-x) \tag{6}$$

Thus we can solve for the end point deflection by the relation[9]:

$$y = \int_0^L \int_0^x \frac{M_x}{EI} du dx \tag{7}$$

Therefore, the designer will now have a good appreciation of how the link will deform during motion. He/she can then design the link to be as rigid or as flexible as desired modifying the stiffness coefficient EI. However, this analysis does not include the reaction moments from adjacent joints. Therefore, this method is only valid for the last link in the robot. To calculate the deflection of all the preceding linkages, the designer must add the maximum reaction moment from the adjacent motor to the moment equation (6). Likewise if gravity is a concern, the respective moment must be added to (6). Since the gravitation moment is dependant on the position of the robot, it is best always to consider the maximum gravitational moment in the analysis.

If a flexible link is being designed, one should note that for an Euler-Bernoulli beam that the model is valid for deflections that are less than 10% the length of the beam. The designer should be aware of all the assumptions/restrictions made in the beam model he/she chooses.

It may be advantageous to use tapered links on a robot. A tapered link can provide an equal amount of strength with a reduction in mass. However, the above deflection analysis is not valid since the second moment of inertia (I) is not constant over the length of the beam. However, it is still possible to calculate the deflection of a linearly tapered link (i.e. a constant linear decrease in I over the length). The solution is obtained from tabulated results such as those given in [10]

Stress Analysis

Once a particular stiffness (EI) is chosen from the deflection analysis, we must then look at the stresses caused by the loading and motion of the link to determine whether permanent deformation will result. We assume that almost all linkages may be considered slender. Therefore the stresses due to the bending moments are paramount. These stresses may be calculated from the relation[11].

$$\sigma_x = \frac{M_z y}{I} \tag{8}$$

Where y is the distance from the neutral axis to the surface of the link. Using the moment equation (6), we can substitute the maximum moment into (8) to calculate the maximum stress in the x direction. Note that the stress in the y direction is zero. Furthermore, the designer should recognize that the link under analysis may experience the accelerations of several joints preceding it. Therefore, the bending stress due to each motion must calculated.

There may also be a torsion applied along the link axis. This will happen if the center of mass of the end effector (of mass m_e) is eccentric(by distance e) from the link axis. Considering, the same constant acceleration $\ddot{\theta}$ from the previous section and the gravitational force, one can calculate the shear stress of the beam from the equation[11]:

$$\tau_{xy} = \frac{(L\ddot{\theta} + g)m_e e c}{J} \tag{9}$$

Where c is largest radius of the cross section and J is the polar moment of inertia. Again, different motions of preceding joints will effect the torsion applied to the link. Therefore calculate the shear stress due to each motion and add the results. Note, take care to note the sign of the shear stress (cw or ccw).

Using the calculated bending stresses and total of the shear stresses applied to the beam we can now combine their effects using Mohr's circle[10]. In doing so, we may calculate the maximum shear stress through the relation.

$$\tau_{MAX} = \sqrt{\left(\frac{\sigma_x - \sigma_y}{2}\right)^2 + \tau_{xy}^2} \tag{10}$$

In almost every case, the beam will most likely fail due to the shear stress exceeding the maximum shear stress theory[11] which states:

$$\tau_{MAX} < 0.5\, S_y \tag{11}$$

where S_y is the maximum yield stress under tension. Therefore, as long as the each combination of bending stress and shear stress yields a combined maximum shear stress which is below the maximum yielding shear stress, then the link will be sufficiently strong.

Natural Frequency Analysis

When designing a link it is extremely important to consider its fundamental natural frequency. For a rigid link, the designer would want a very high fundamental frequency. However on the opposite extreme, a flexible beam may be desired to have a specific low fundamental frequency. For the latter case if we consider a cantilever beam of negligible mass with an end point mass m which deflects under gravity, an estimate of the natural frequency is given by[12]:

$$f_g = \frac{50}{\sqrt{\delta_L}}\ [Hz] \qquad (12)$$

where δ_L is the static deflection due to gravity and is measured in meters. The static deflection may be calculated through the equation[12]:

$$\delta_L = \frac{mgL^3}{3EI}\ [m] \qquad (13)$$

Note, this approximation technique is valid only for a flexible beam where the mass is negligible compared to the end mass.

For the case where the mass of the link may not be neglected, the relationship for the natural frequency becomes much more complicated. It is suggested to the reader that a good reference to vibrational analysis is [13] if he/she wants to pursue the topic.

As a general note, the above calculations for the deflection, strength and natural frequency of a joint will become increasing more difficult for robots with many degrees of freedom. Rather than compute these characteristics in some small simulation package that one could program, we suggest the use of a solid modelling package such as IDEAS. A solid modelling software will calculate all of these linkage properties for a given model with dynamic simulation and finite element packages. Since these modelling programs are becoming easier to use, we suggest their use even for the simplest link.

Material Selection

In all of the above analysis, the two recurring material properties are the Young's modulus (E) and the density. It would be best to select a material which maximizes the Youngs modulus without providing a larger increase in the density. However, although there has been great improvements in the field of metallurgy and material sciences, there has been no recent development which increases the Youngs modulus of common materials such as steel and aluminum[8]. In comparison, these two materials are very close in performance. The ratio of Young's modulus over the density is 2.7 and 2.6 for steel and aluminum respectively. However, aluminum is much more ductile than steel and has a low fatigue resistance. Therefore we suggest the use of a common alloyed steel for the construction of the link.

There are light weight materials such as titanium and beryllium which will outperform steel or aluminum however their high cost precludes their use. Furthermore in all the previous analysis, it was assumed that the Young's modulus was linear in the elastic zone. This stipulation neglects the use of plastics and carbon fibre composites for a linkage since these materials have a non-linear relationship between stress and strain. Moreover, these plastic composite materials tend to exhibit creep over prolonged constant loads such as gravity. However, plastic materials do exhibit a much higher damping than metallic materials which is highly desirable.

Joint Design

It is difficult to list an algorithm for the design of a robotic joint since the geometry can vary from one application to another. Therefore in this section we will list the qualities of a good joint design and leave the actual implementation to the designer.

i) The joint must be sufficiently stiff for any combination of the largest moments and forces applied to it from the motor, dynamic response and gravitational loading. This strength analysis is static making the strength calculations easier.

ii) The motor output shaft should be sufficiently supported by bearings to absorb all axial or radial loads and any moments. The use of angular contact bearings will guarantee such support if they are rated for the load applied to the shaft.

iii) Make the design modular so that some standard connection may be made between the link and joint. It is a good idea to use bosses and dowels to ensure that the links will attach to the joint at one specific direction and orientation.

iv) Design any critical drive components to high tolerances to prevent relative motion between parts. For example, the tolerance of keyway must be such that a light pressfit is required. This type of design will eliminate any "slop" in the transmission.

v) Lastly, make sure that the joint will be able to be assembled upon fabrication. It is quite easy to design the most perfect joint only to discover that there is no possible way it can be assembled. Furthermore, a designer must take into consideration the manufacturing limits for the production of all the parts.

Power Supplies

The selection of the power supply is relatively straight forward once the appropriate motors have been selected from the dynamic analysis. A designer must find a power supply that can provide the motor with its specified power consumption [Watts] and type (AC or DC). However, as was discussed previously, the DC motor is the best option for providing linear control of either the speed or the torque. Therefore we are assuming that a DC power supply is being used for the rest of this section.

The power supply must also be capable of producing both the maximum voltage and current that the motor commands. Note that the peak current and voltage for the motor will never occur simultaneously (refer to motor specifications). Furthermore the designer must make sure that the location of the robot must have a power source that the power supply can use. For example the consideration of the number of phases (single or three) and the maximum voltage and current of the power source is a concern when selecting the power supply. Lastly, the voltage of the power supply should be regulated to avoid any voltage fluctuations from the source.

Similarly to the selection of the power supply, a servo amplifier must be chosen so that it too meets the power, voltage and current constraints of the motor. Also the amplifier should be linear for ease of modelling and so that linear control theory can be applied. The type of amplifier will depend upon whether torque or speed control is to be used. For torque control, we would need a current amplifier and for speed control we would need a voltage amplifier.

Sensors

To control the robot for an experiment (either position or force), a designer will need some form of measurement to feedback and close the control loop. This feedback is provided through measurements from sensors. The type of measurement will depend on the type of control the designer wants.

Position Control

For position control, the motor angular position must be used in the feedback. To measure the position there are two alternatives i)encoders or ii)potentiometers.

An encoder is an optical device which produces a digital signal that corresponds to angular motion. Each change in state of the signal means a discrete amount of angular motion has occurred. The total rotation of the shaft for a finite period of time can be determined by counting all the changes in state of the encoder signal. Furthermore, by a special signal processing technique called quadrature, two encoder signals which are 90 degrees out of phase can be processed to increase the resolution by 4 and provide the direction of the rotation. The resolution of the output rotation is incredibly small if a large reducer (such as a harmonic drive) is used. For example, consider an encoder of 500 counts per revolution and a harmonic drive with reduction ratio of 100:1. Each count of the encoder would represent a rotation of:

$$Resolution = \frac{360}{4*500*100} = 0.0018 \left[\frac{deg}{count}\right] \tag{14}$$

of the output. However, The problem with encoders are that they are incremental meaning that they measure angular displacement relative to a starting position (ie. there is no absolute measurement). This shortcoming of encoders may be avoided by homing the robot to a specific position before each experiment.

A second method to measure angular position is through a linear rotary potentiometer attached to the motor shaft. As the shaft rotates, the voltage over the potentiometer will increase or decrease linearly depending on the direction of rotation. Thus there is an infinite resolution in theory, but realistically the resolution will be determined by the resolution of the analog to digital converter(refer to the Data Acquisition section). However, there are two drawbacks to using potentiometers. Firstly, similar to encoders, the potentiometers are incremental sensors and thus must be homed before each experiment. Secondly, there will be a fixed number of rotations that a potentiometer may travel. If the designer wants a 360 degree motion, the potentiometer must have twice the number of rotations as the reduction ratio of the reducer to allow full rotation in either direction.

Velocity Control

It is often desired to have a velocity feedback to a controller. There are two most common methods for measuring the velocity of a robotic joint. Firstly, the most traditional method is to attach a tachometer to the motor shaft. A tachometer produces a voltage signal which is proportional to the angular speed of the motor. The second method is to differentiate an encoder signal with respect to time using a first order backwards difference[14]:

$$\dot{\theta} = \frac{\theta_i - \theta_{i-1}}{T} \tag{15}$$

where T is the sampling period. We suggest the use of the latter approach to measure the angular velocity because then only one sensor (the encoder) is needed for both position and velocity measurements. Note that since the measurement of the angles in (15) is discrete, the error for the velocity measurement will increase if the sample period T decreases.

Force Control

There are two options for sensor feedback to control the force that an end effector applies to a surface. Firstly, the output torque of each joint may be measured and used to calculate the applied force. The torque output of a joint could easily be determined by measuring the strain deformation of the flex spline in a harmonic drive. The relationship between the strain deformation and the output torque is linear for a special configuration as detailed in[8]. If a harmonic drive is not being used, then some form of compliant structure must be attached to the output and have its deformation measured. The other alternative is to place a force transducer at the end effector to directly measure the applied force. The size of the transducer may be easily determined by calculating the reaction force required to restrain the robot when all the motors apply their maximum torque at their largest moment arm. Although this may be a more direct approach to measuring the endpoint force, it offers less protection for the transducer which is typically easily damaged if overloaded. Since the main focus of this robotic design is for experimentation where the outcome is never exactly known, the former approach to force measurement offers more protection for the sensor.

Estimating the Deflection and Orientation of a Flexible Link

When a flexible link of a robot deforms under a load, the position and orientation of all subsequent joints and links are unknown. Therefore it is of paramount importance that the orientation and position of a flexible link be measured or estimated. To measure these variables there is a choice of two approaches. Firstly, a vision system may be used to estimate the position of the endpoint[15]. However, this method is very indirect and has many factors such as lighting, blockage of the endpoint etc. which will effect the estimation. A more direct approach is to measure a finite set of curvatures along the beam length and estimate the position and orientation of the end point with these readings[16]. The beam curvatures may be measured directly through the use of strain gauges in a half bridge configuration by the relation:

$$\frac{d^2y}{dx^2} = \frac{4}{V_{ex}K_g Gt} V_{out} \tag{16}$$

where K_g is the gauge factor of the strain gauges, V_{ex} is the voltage excitation of the bridge, t is the thickness of the beam and G is the gain of the strain gauge amplifier. This latter method is less susceptible to external disturbances than the vision system thus giving it a more robust performance.

Data Acquisition and Control

The last stage of design for a robotic system is in choosing the data acquisition and control system. Presently, there are two practical choices of computers to run the system on i) personal computer (IBM,Mac) ii) a work station (HP 720, Sun).

Consider the use of a work station to control a robot. These computers have large amounts of processing power and have a standard VME bus which communicates all of the input and output signals. However, these computers are extremely expensive. Generally a researcher could not afford such a system in the initial development of a robotic laboratory.

The alternative is to use a PC based data acquisition and control system. It is immediately apparent that such a system lacks computational power. However, if the robotic systems which are designed are simple (3 DOF or less), then a PC's performance would be adequate. Furthermore, the power of the PC can be enhanced by using an EISA bus, direct memory access(DMA) and transputers. There are processing boards that can provide up to 25 Mflops of computational power[17]. Note that even with all of these communication and processing enhancements, the PC based control system is still less expensive than the work station.

It is also the responsibility of the designer to ensure that all the sensor signals can be input and the control signals output by the computer. There are generally three types of data input/output for a PC. These are analog to digital(A/D), digital input/output (DIO) and digital to analog outputs (D/A).

The number of D/A channels will be equivalent to the number of servo amplifiers used. The number of A/D channels needed for the system will be determined by the number of tachometer, potentiometer, transducer and strain gauge signals used. In the case where a large number of signals are to be measured, it would be more economical to multiplex one A/D channel for the entire group. Lastly the number of digital ports will be reflective of the number of encoders used. Note that there are some commercially available boards which will process and communicate encoder signals through an expansion slot in a PC.

Lastly the resolution of the A/D and D/A channels should be sufficient to provide a fine discretation of the measured (or output) voltages. An A/D or D/A channel with 12 bits of resolution will provide a measurement (output) in discrete levels of 4.88 mV when operating over the bipolar range of 10 Volts. This is usually a satisfactory for most robotic measurements(outputs).

The controller of a robot may be digitally implemented through programs in a structured language like C which will allow the programming of interrupt service routines. Such an implementation allows very easy modification (or replacement) of the controller.

Conclusions

An iterative design procedure has been discussed which allows a designer to create a robot from initial conception to final product following the chronological steps of i) robotic mechanics ii) power supply iii) sensors and iv) data acquisition and control. The first stage of the design is by far the most important for it will give the designer an appreciation of the size of the robot and the necessary power needed to manipulate it. Furthermore in this stage a designer may quickly come to terms with the limitations his/her robot and may then quickly modify the experiment or design constraints to meets these limitations.

We have successfully used this technique for the design of three robotic systems for experimentation. We are confident that if this procedure is followed that the desired robot will be produced and customized to the researchers needs.

References

1. M.W. Spong, M. Vidyasagar "Robot Dynamics and Control", Wiley & Sons 1989
2. B. Armstrong, "Friction: Experimental Determination, Modeling and Compensation", IEEE Conf. on Robotics and Automation, Philidelphia April, 1988
3. "Harmonic Drive Gearing: CUP Type HDUC and HIUC Component Sets", specification catalogue from HD Sytems (manufacturer of harmonic drives)
4. T. Wildi, "Electrical Power Technology", Wiley and Sons, 1981
5. B.C. Kuo, "Automatic Control Systems 6th Edition", Prentice Hall, 1991
6. C.L. Phillips, R.D. Harbor, "Feedback Control Systems 2nd Edition", Prentice Hall, 1991
7. M. Hashimoto, K. Koreyeda, T. Shimono, H. Tanaka, Y. Kiyosawa, H. Hirabayashi, "Experimental Study on Torque Control Using Harmonic Drive Built-in Torque Sensors", Proc of Int. Conf. on Robotics and Automation, Nice, France, May 1992
8. E.I. Rivin, "Mechanical Design of Robots", McGraw Hill, 1988
9. R.J. Roark, W.C. Young, "Formulas for Stress and Strain, 5th Edition", McGraw Hill, 1982
10. F.P. Beer, E.R. Johnston Jr., "Mechanics of Materials", McGraw Hill, 1985
11. J.E. Shigley, C.R. Mischke, "Mechanical Engineering Design 5th Edition", McGraw Hill, 1989
12. A.D. Christian, W.P. Seering, "Design Considerations for an Earth Based Flexible Robotic System", IEEE International Conf. on Robotics and Automation, 1989
13. L. Meriovitch, "Analytical Methods in Vibration", Collier Macmillian Ltd., London
14. P.R. Belanger, "Estimation of Angular Velocity and Acceleration from Shaft Encoder

Measurements", IEEE Proceedings of International Conf. on Robotics and Automation, Nice, France, 1992

15. C. Chen, Y.F. Zheng, "Deformation Identification and Estimation of One-Dimensional Objects by Vision Sensors", Journal of Robotic Systems 9(5), 1992

16. S.J. Miller, J.C.Piedbouef, "Estimation of Endpoint Position and Orientation of a Flexible Link Using Strain Gauges", Proceedings in 1993 DND WorkShop on Advanced Technologies in Knowledge Based Systems and Robotics, Ottawa

17. 1992 IEEE-488 and VXIbus Control, Data Acquisition and Analysis Catalogue, National Instruments Ltd.

Y03246

A Practical Dynamic Model of a Puma 560 Manipulator for Aircraft Maintenance

Gordon Hardy
Defence Research Establishment Pacific
FMO Victoria, B.C. V0S 1B0, Canada

#149721

Abstract

Manipulator arms performing aircraft maintenance must have high positioning and orientation accuracy. External position sensors may be necessary to achieve this and a dynamic model of the manipulator is required to use the sensor information optimally. This paper describes a system-identification-based method to determine a computationally efficient dynamic model for a PUMA 560 manipulator that can be easily adapted to other manipulators. The resulting model consists of a sample and hold, a delay, and a plant. The model was derived by fitting standard first- and second-order functions to responses measured at a variety of positions, speeds and motion directions. Statistical analysis of the model coefficients indicated that the manipulator's steady state gain is sensitive to speed and direction. In addition, it showed that radial and vertical motion can be modeled adequately with a first-order plant, but tangential motion requires a second-order plant. When subjected to a white noise input, the model estimated the measured position with a root mean square error of 0.8 mm.

Introduction

Background

Currently, aircraft such as the CF-18 are manually inspected for defects by scanning suspected areas with ultrasonic or magnetic sensors. A PUMA 560 manipulator was used at the Defence Research Establishment Pacific (DREP) to demonstrate the feasibility of using a robotic based scanner to perform non-destructive aircraft inspection. Such a scanner has the advantages of producing repeatable results and reducing tedious, error-prone work. Inspection devices must be accurate enough to repeatedly detect delaminations 1 mm long in composite structures, and to keep the ultrasonic transducer normal to the surface. For most manipulators, including the PUMA, external position sensors are required to achieve these criteria. However, a dynamic model of the manipulator is required to optimally use these sensors' data.

Existing Dynamic Models

One approach to modeling a robot's dynamic behavior, is to model its individual components, the control algorithm and the physical model. The control algorithm receives a command to go to a position and uses sensor information to direct the joint motors. The physical model relates the motor torques to the joint accelerations.

The control algorithm first translates the position command to joint angles by solving the inverse kinematic problem. The solution requires 24 parameters including joint lengths, linear offsets and angular offsets. These parameters can be difficult to obtain for a specific robot arm. The control algorithm employs angular position and velocity sensors to drive the joint motors to the joint angle vector solution. To make matters worse the control algorithm is typically not published

Physical models are also difficult to implement. Fu[1] described three conventional physical models: Lagrange-Euler, Newton Euler and generalized D'Alembert. Unfortunately, these models are either computationally slow or complex to derive. Their greatest problem, however, is that they require measurements of mass, moments of inertia and length for each manipulator link. To illustrate the difficulty, Armstrong[2] reported having to disassemble an arm to measure these properties.

A Practical Approach

A more practical and generic method uses system identification theory to determine the transfer functions relating the requested-position input to the end-effector-position output. This method does not require knowledge of the internal control algorithms or the physical parameters of the manipulator. Bossert[3] described a method to determine the model of a PUMA 560 manipulator based on system identification theory. He related torque to angular position for each joint. Whereas in this paper, the position command is related directly to the end-effector position, thus treating the entire manipulator as a black box. The method involves measuring manipulator responses and fitting standard step-responses to them. First-order transfer function coefficients include: steady state gain (K), time delay (T_d), and the time constant (τ). Second-order coefficients include steady state gain, time delay, natural frequency (ω) and the damping factor (ζ). Analysis of variance was used to determine the dependency of the model coefficients to speed, position and direction of motion. The most dependent relationships were then incorporated into the model.

PUMA 560 "ALTER" Mode

Like many manipulators, the PUMA 560 has a facility to use external sensor information. Normally, user written programs direct robotic motion from one position to another. In "ALTER" mode, the PUMA receives position correction commands to direct motion, which must come every 28 ms from an external computer ("external ALTER") or an internal task ("internal ALTER"). The method described in this report can be applied to any manipulator that accepts position commands in a similar way.

The System Identification Approach

A manipulator can be represented as a linear model (Fig. 1) for each of three directions in a cylindrical coordinate system: radial or x-direction, to and from the robot base, tangential or y-direction, and vertical or z-direction. When operating in "ALTER" mode the PUMA 560 manipulator receives a motion command every 28 ms, which is modeled by a sample and hold transfer function

$$G_H(s) = \frac{1 - e^{-Ts}}{s} \qquad (1)$$

where T - update period = 28 ms

s - Laplace domain independent variable.

After receiving the command, the arm jumps to the destination. This action is modeled by a plant, P, and a delay, $G_D(s)$.

$$G_D(s) = e^{-sT_d} \qquad (2)$$

where T_d - time delay (s).

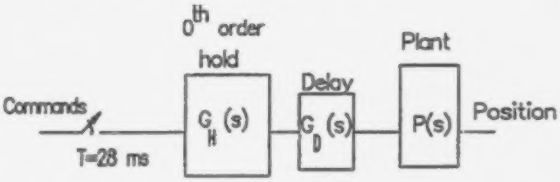


FIGURE 1: Block diagram of "ALTER" motion.

Identifying the Plant

Like the impulse response, the step response can identify a plant, but unlike the impulse response, the step response is more practical to measure. In this case, a step response is a recording of the end-effector position when the arm receives a position command.

Step Response Models

The measured step response of the manipulator is fitted to standard first- and second-order step responses. The first-order plant representation is

$$H(s) = \frac{K}{1 + \tau s} \qquad (3)$$

where K - steady state gain

τ - time constant

H - plant representation.

Multiplying the above equation by a step function, and translating to the time domain gives the first-order step response

$$c(t) = K\,(1 - e^{-t/\tau}). \qquad (4)$$

The standard second-order representation is

$$H(s) = K \frac{\omega^2}{s^2 + 2\zeta\omega s + \omega^2} \qquad (5)$$

where H(s) - second-order transfer function

K - steady state gain
ω - natural frequency
ζ - damping factor.

The second-order step response depends on whether the system is over damped or under damped. If over damped ($\zeta>1$), the step response is [4]

$$c(t) = K + \frac{K\omega}{2\sqrt{\zeta^2 - 1}}\left(\frac{e^{-s_1 t}}{s_1} - \frac{e^{-s_2 t}}{s_2}\right), \quad (6)$$

where $s_1 = \omega\left(\zeta + \sqrt{\zeta^2 - 1}\right)$ (7)

$$s_2 = \omega\left(\zeta - \sqrt{\zeta^2 - 1}\right). \quad (8)$$

If under damped ($\zeta<1$), the step response is [4]

$$c(t) = K - \frac{Ke^{-\zeta\omega t}}{\sqrt{1 - \zeta^2}} * \sin\left(\omega t\sqrt{1 - \zeta^2} + \tan^{-1}\frac{\sqrt{1 - \zeta^2}}{\zeta}\right). \quad (9)$$

Measuring the Step Responses

Configuration

Fig. 2 shows the test bed used to measure the step responses for a PUMA 560 manipulator. All measurements were taken with no end-effector load. The arm was oriented so that the radial axis corresponded to the puma's internal y-axis.

Measuring position as a function of time

The response to a step input was measured using a toothed wheel connected to a potentiometer, as shown in Fig. 3. A wooden platform structure was placed in contact with the wheel so that when the arm moved, the wheel would rotate. The resulting voltage change was recorded on an oscilloscope. A photograph of the oscilloscope traces was manually digitized. To measure time delay, the program toggled a digital output to indicate on the oscilloscope when the step command was given.

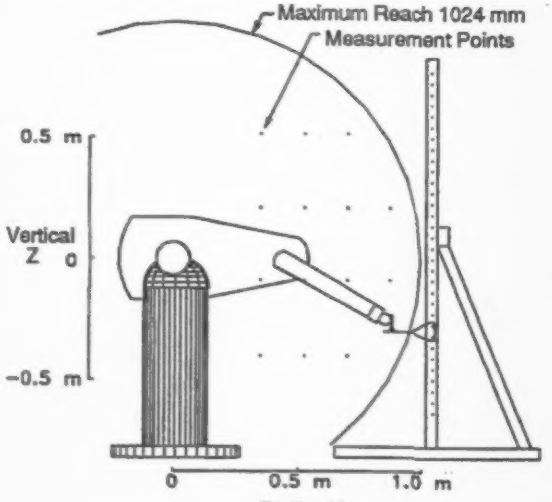


FIGURE 2: Manipulator test bed.

VAL II program

The essential parts of the program producing a step in the x-direction are listed in Appendix A. It was written in a BASIC-like robotic language called VAL II, running on the PUMA 560 controller. It consists of two routines: "move", a routine interrogating the user and initializing the robot; and "jumpx.pc", a background task producing a step command synchronous to the internal 28 ms clock.

Measurement Parameters Considered

Three test parameters that may affect the dynamic response of the manipulator were considered. Position and motion directions were considered because they involve different inertias which change the dynamic model. Speed was considered because the arm response may differ with speed. Taking measurements throughout the entire work volume was unnecessary, since the arm inertias remain constant with rotation around the vertical axis. Consequently, only 14 positions within a vertical slice of the volume were examined (see Fig. 2). Tests were done at four different speeds (0.071, 0.14, 0.28 and 0.68 m/s) and three different motion directions (tangential, radial, and vertical). Fig. 2 defines two of the three axes.

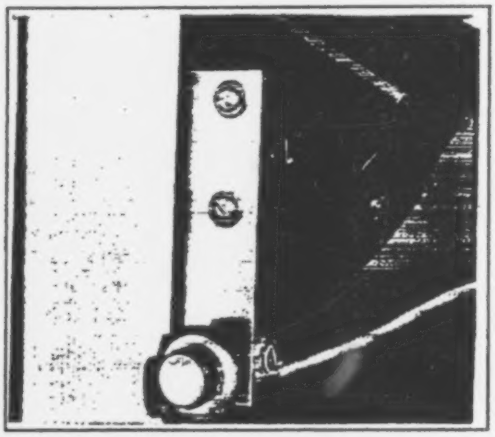


FIGURE 3: End effector.

Determining and Combining Model Coefficients

Using Minimization to Determine the Model Coefficients

A trial was run for 86 combinations of position, speed and direction. For each trial, the manipulator response was measured and fitted with a standard first- and second-order step response. Each solution set includes five model coefficients: delay, steady state gain, time constant τ, natural frequency ω, damping factor ζ. Time delay and gain were measured directly. The other terms were determined by minimizing a root mean square (RMS) error function that compared the measured step responses to a standard first- and second-order step response model using the program MATLAB[5].

$$\text{RMS error} = \sqrt{\frac{\sum_{i=1}^{N}\sum_{k=1}^{K}\left(\mathrm{R}_i(\mathrm{T}_s k) - M(\mathrm{T}_s k)\right)^2}{N \cdot K}} \tag{10}$$

where, T_s - sampling period = 5.63 ms

K - number of samples = 30

M - model position predicted

- First-order model : Equation 4

- Second-order model : Equation 6 or 9

R - measured step response

N - number of step responses.

Starting with an initial estimate, MATLAB's minimization algorithm varied the model coefficients to converge the error function to its minimum.

Local minima

A major concern with using a minimization algorithm is the danger of converging to a local minimum. Plotting the error function as a function of the model coefficients can indicate potential problems. In many cases, such plots indicate a unique solution. However, in some cases multiple solutions were observed. For example, Fig. 4 shows contours of the RMS error function as a function of natural frequency and damping factor. The error function surface is smooth, indicating no isolated local minima. However, the surface has a long flat minimum, indicated by the dotted line segment. Cases with this structure have multiple second-order solutions, indicating that a first-order solution may suffice (corroborated in the section entitled "Graphical Method").

Combining the Solution Sets

To create a practical model, the 86 solution sets must be combined into one model. Not all the coefficients will vary significantly as the test parameters change. Some coefficients may vary for different positions, speeds or motion directions, and therefore, have to be accounted for in the final model. Other coefficients may prove to be independent of the parameters, and can be treated as constants. Analysis of variance[6] can quantify the dependency of the coefficients on the experimental parameters, providing that the coefficient values are normally distributed.

Normality of the coefficients

Coefficient histograms compiled from the solution sets showed which were normally distributed. Histograms for ω, ζ and RMS errors were clearly not bell shaped, and therefore not normally distributed. Fig. 5 shows histograms of the remaining coefficients considered to have a near normal distribution.

Graphical method

The sensitivity of the non-normally distributed coefficients, ω and ζ, and the RMS errors, can be tested graphically. Fig. 6 shows the dependency of the RMS errors. Each subplot was composed by sorting the solution sets from all 86 trials into groups according to the subplot's corresponding parameter. The range of values for each group is illustrated as a two-standard-deviation error bar. The RMS error values were used as a measure of how well the responses fit standard step functions. Fig. 6a shows that to ensure an accuracy of less than 1 mm, the final model should only be used for y-positions less than 700 mm. Fig. 6d and 6h indicate that tangential motion is best modeled with a second order plant. Graphs for ω and ζ were not included because their values approached infinity, consequently their averages and graphs were meaningless.

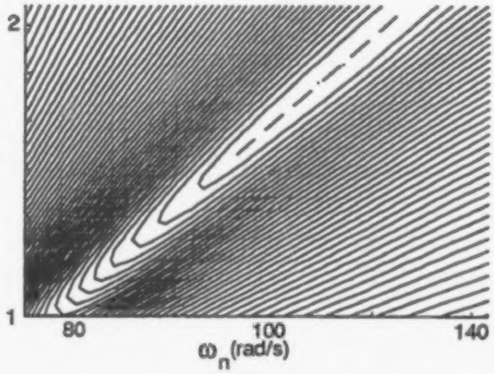


FIGURE 4: A contour plot of a second-order error function with multiple solutions.

Statistical method using analysis of variance

Analysis of variance[6] tests how sensitive the normally distributed coefficients are to the experimental parameters. The results in Table I are expressed as an i-ratio (the ratio of the result to the value in the 95%-confidence f-distribution table[6]). If the ratio is less than one, the means are statistically the same and consequently the corresponding coefficient and parameter are independent.

The final model should first account for the relationships with the greatest dependencies (highest i-ratio). Table I shows that the steady-state gain is highly dependent on motion direction; consequently, the model includes a separate equation for each direction. Table I also shows that the gain is dependent on speed and consequently, the gain is expressed as a linear function of speed. Other dependent relationships are not incorporated in the model, because they would greatly increase model complexity with only a small increase in accuracy. Since τ is independent of all parameters, it is regarded as a constant.

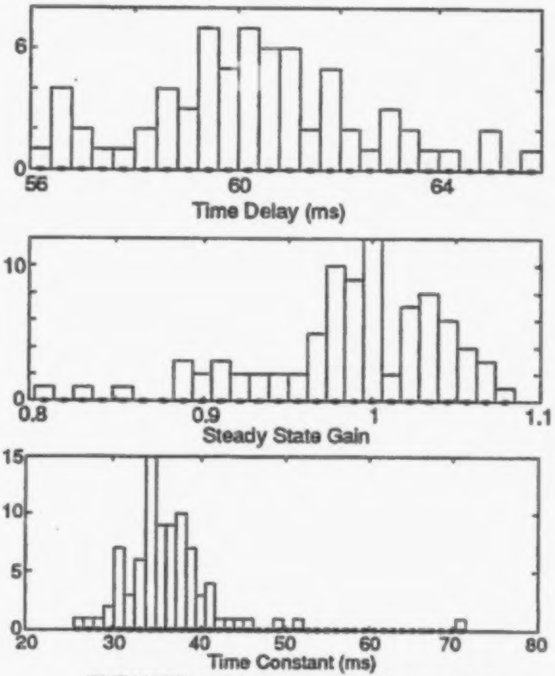


FIGURE 5: Histograms of the model coefficients.

The Combined Model

The resulting model consists of a transfer function for each of the three directions along the axes of a cylindrical coordinate system. Each transfer function, shown in Fig. 1, consists of a 28-ms sample and hold, a delay, and a plant.

Delay

"Internal ALTER" mode

For the program listed in Appendix A, the average measured delay was 60.45 ms. This figure greatly depends on what instructions are executed between the "WAIT" command, which waits for an internal 28 ms clock transition, and the "ALTOUT" command, which starts motion.

"External ALTER" mode

In "external ALTER" mode, the delay from when the puma requests data to when it starts moving is 38.5 ms. A complete discussion appears in the manual[7].

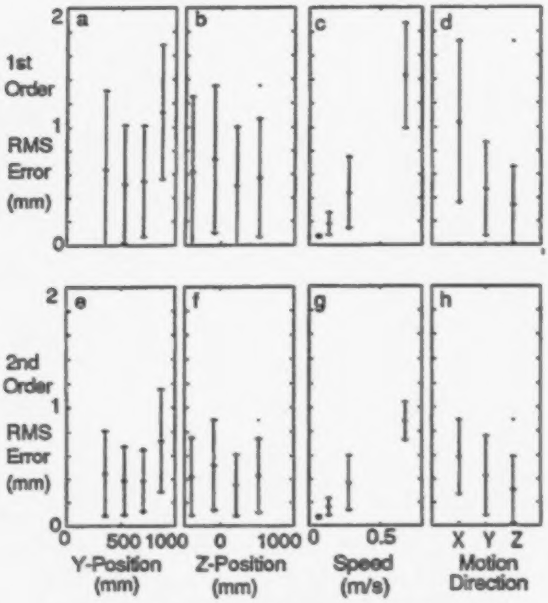


FIGURE 6: Dependence of RMS error coefficients to position, speed and direction of motion.

The Plant

From 68 trials, a combined plant was created and is summarized in Table II. Steady state gain is expressed as a linear function of speed.

$$k = a\,s + b \tag{11}$$

where, k - steady state gain

s - speed (in m/s)

a,b - constants.

Motion in the radial and vertical direction is modeled with a first-order plant (Equation 3). The time constants are the average for all trials in the appropriate direction.

Motion in the tangential direction is best modeled with a second-order plant (Equation 5). Since ζ and ω approach infinity, an average value would produce a poorly fitting model. Consequently, another minimization of the error function (Equation 10) was performed using data from all the tangential trials.

TABLE II: Combined model coefficients

direction	a (m/s)	b	ζ	ω (rad/s)	τ (ms)
x (tangential)	0.0518	1.010	0.864	53.9	
y (radial)	0.0818	0.933			42.5
z (vertical)	0.134	0.941			35.6

Model Performance

The combined-model RMS error was calculated for each trial. The results were grouped according to their y-position. The mean and standard deviation of

TABLE I: Model coefficients sensitivity to measurement parameters

		delay (ms)			steady state gain			τ (ms)		
parameter	value	mean	standard deviation	i-ratio	mean	standard deviation	i-ratio	mean	standard deviation	i-ratio
speed	0.14 m/s	60.4	2.5	0.52[a]	0.95	0.06	4.57	38.0	4.8	0.11[a]
	0.28 m/s	60.0	1.9		1.00	0.04		36.3	6.8	
	0.68 m/s	61.3	2.2		1.02	0.03		39.0	22.5	
direction	x	62.1	1.7	1.58	1.04	0.028	7.44	34.9	3.5	0.37[a]
	y	60.4	2.1		0.96	0.058		42.6	20.5	
	z	59.9	2.1		0.97	0.042		35.6	4.2	
y[b] position	350 mm	61.3	2.3	0.73[a]	0.99	0.058	0.37[a]	34.5	3.3	1.11
	525 mm	60.0	2.0		1.00	0.044		35.8	4.0	
	700 mm	60.3	1.5		0.97	0.066		37.6	5.1	
z position	-400 mm	61.7	2.0	1.63	0.96	0.064	1.12	34.7	5.5	0.42[a]
	-90 mm	59.5	2.2		1.00	0.042		41.2	21.5	
	210 mm	60.8	2.0		0.99	0.061		37.0	4.3	
	515 mm	59.8	1.5		1.01	0.042		36.7	3.4	

[a] Considered independent

[b] 870 mm absent because it is greater than the 700-mm model constraint

each group are compared in Fig. 7a. It shows that for 1 mm accuracy, the model should be used for y-positions less than 700 mm. The procedure was repeated for the remaining three parameters. Fig. 7c shows that model error is approximately proportional to speed and for 1 mm accuracy, the speed should be less than 0.5 m/s

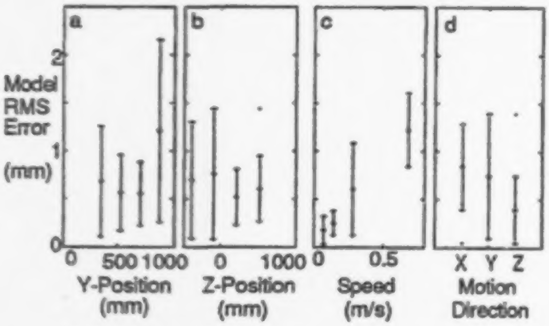


FIGURE 7. Model error verses the four parameters grouped by position, speed and direction of motion.

Model Verification Using White Noise

Most models can be verified by comparing the responses of the simulated and the actual systems to a white noise input[8]. The chosen white-noise signal, shown in Fig. 8, is a binary random signal that switches between states with a 50% probability every 28 ms.

Figs. 8 and 9 summarize the simulated and actual responses to tangential, radial and vertical motion respectively. Measurements were taken at a single point (Y=700 mm, Z=-90 mm) using the same position sensor shown in Fig. 3. A digital storage oscilloscope recorded the input and the measured position, with 8-bit resolution every 4 ms, and transferred them to two data files. A MATLAB[5] routine read the input data, fed the result into the model, and compared the output to the measured position. To compare the input and output, a bias was applied to give them an average value of zero[8].

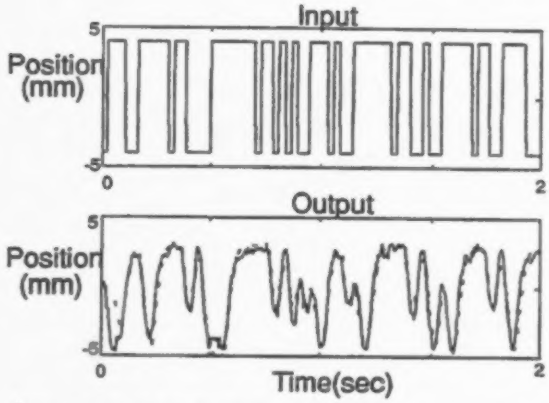


FIGURE 8: Model (dashed line) and measured (solid line) response to white noise in tangential direction.

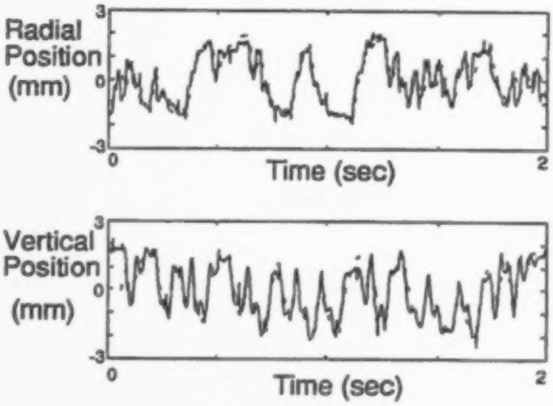


FIGURE 9: Model (dashed line) and measured (solid line) response to white noise in radial and vertical direction.

Table III tabulates the RMS error for the data shown in Figs. 8 and 9. The tabulated data are well within the predicted model errors of Fig. 7.

The graphs and the table clearly indicate that the model accurately predicts the measured position, although not necessarily the actual position, and that the method used to derive the model is trustworthy.

TABLE III: White noise RMS error

Direction:	X (Tangential)	Y (Radial)	Z (Vertical)
RMS error (mm):	0.67	0.35	0.41

Conclusion

Significance

Manipulator arms performing aircraft maintenance must have high positioning and orientation accuracy. External position sensors may be necessary to achieve this criteria, but they require a dynamic model of the manipulator to use the sensor information optimally. This paper described a system-identification-based method to determine a computationally-efficient dynamic model for a PUMA 560 manipulator. The method can be applied to any manipulator with an operating mode like PUMA's "ALTER" mode.

The Model

The resulting PUMA 560 model consists of a transfer function for each of the three directions along the axes of a cylindrical coordinate system. Each transfer function consists of a 28-ms sample and hold, a delay, and a plant. The time delay is 60.45 ms for internally generated path corrections. The plant was derived by fitting standard first- and second-order step responses to step responses measured for different parameters (position, speed, and direction). Analysis of variance showed that the most sensitive model coefficient, steady state gain, is dependent on motion direction and speed. A first-order plant is capable of modeling radial and vertical motion, but a second-order plant is needed for tangential motion.

Performance

The resulting model worked best for radial distances less than 700 mm from the robot base and end-effector speeds less than 0.5 m/s. The root-mean-square position error of the model is roughly proportional to the end-effector speed, and has an average value of 0.8 mm.

When subjected to white noise, a simulation of the model tracked the actual measured position within the predicted error. The white noise tests verified that the model is sufficiently accurate and the method trustworthy.

References

1. K.S. Fu, R. Gonzalez, C. Lee, "Robotics: Control, Sensing, Vision and Intelligence",McGraw Hill, Toronto, 1987.
2. B. Armstrong, B. Khatib, O. Khatib, J. Burdick, "The Explicit Dynamic Model And Inertial Parameters Of The Puma 560 Arm", IEEE International Conference On Robotics And Automation, 1, Pg 510-18, 1986.
3. D. Bossert, G.Lamont, M. Leahy, I. Horowitz, "Model-based Control with Quantitative Feedback Theory; Empirical Model Analysis",Proc. of the 29th IEEE Conference on Decision and Control, Pg 1974-5, 1990.
4. K. Ogata, "Modern Control Engineering", Prentice Hall Inc., Toronto, 1970.
5. Computer code MATLAB (The Math Works Inc., South Natick, M.A., 1990)
6. R. Walpole, R. Myers, "Probability And Statistics For Engineers And Scientists", Macmillian Publishers Co. Inc., New York, 1978.
7. "User's Guide to VAL II", Unimation, A Westinghouse Company, Pittsburgh, 1986.
8. L. Ljung, "System Identification, Theory for the User", Prentice-Hall Inc., Toronto, 1987.

Appendix A: VAL Program Listings

```
.PROGRAM move
        ...
        ALTER (-1, 18)  ;non-cumulative world
         coordinates, alter enabled
        PCEXECUTE jumpx.pc, -1, 0
        ...
.PROGRAM jumpx.pc
        x = 0
        angle = angle+step
        IF angle >= 10 THEN         ; if triggered
          x = a                     ;step size
          SIGNAL -17
        ELSE
          SIGNAL 17
        END
        TYPE /N, x, angle           ;print to queue
         ALTOUT 0, x*TODIS
         WAIT                       ;until 28 ms
```

Computational Aspects of Real-Time Robot Dynamics Simulation

Xavier Cyril, Michael Huculak, Rick Megley
Space Systems Engineering, CAE Electronics Ltd.
C.P. 1800, St.Laurent, Quebec, CANADA H4L 4X4

#149722

Abstract

The real-time simulation of robot dynamics has received considerable attention in recent years. Such simulations support the design and development of robotic manipulators, and form the core of the simulators for training and human-machine-interface studies.

Dynamics models of robots are non-linear and computationally intensive, hence their numerical simulation in real-time is a formidable task. The two major steps involved in the dynamics simulation are: 1) the formulation of the equations of motion (dynamics model); 2) their numerical solution (linear equation solution and numerical integration). Traditionally, the first step is formulated recursively, whereas, the second step may either be tackled recursively or non-recursively. The essential difference between the two is that only the recursive method is algorithmic, in the sense that it follows a procedure that performs calculations at the k-th step using quantities that were calculated at the (k-1)'st step. While recursive methods lend themselves to efficient implementation on uni-processors, non-recursive methods take advantage of multi-processors through vector processing and parallel processing.

Schemes to optimize/customize the implementation of these algorithms have been published in the literature, which range from pipeline processing to specialized VLSI chips. Complicated by the yet unsolved problem of appropriate job distribution and message passing among different processors and the overall analysis of time efficiency, the parallel processing of dynamics algorithms has achieved little success. The key to success in tackling the computational burden lies in understanding the rate of change of the different variables involved in the dynamics simulation process. Generally, the coefficients of the equations of motion do not change at the same rate as the state variables for a robot manipulator. Thus, parts of the dynamics simulation algorithm can be computed at a slower rate than others. This simplifying assumption coupled with the possibility that the formulation and solution of the equations of motion could be computed in parallel leads to the realization of a real-time simulator. A brief review of the published methodologies to obtain and solve the equations of motion is presented. Also presented is a discussion on the dynamics simulation approach mentioned above and our experience in resolving the problems associated with it.

1 Introduction

Dynamics simulation plays a crucial role in the mechanical design of robotic manipulators and for manipulation task training of personnel. This activity is hindered by the complexity of the dynamics itself. The computational complexity in manipulator dynamics has been studied in the last few years with respect to various different approaches. Hollerbach [7] did a comparative study of computational complexity as applied to different methods of dynamics formulation and presented what was then an efficient recursive Euler-Lagrange (EL) formulation. Since then, it has been noted [22] that the recursive EL formulation and the recursive Newton-Euler (NE) formulation [17], [24], [19], [12], [13], are computationally equivalent and, obviously, more efficient than their non-recursive counterparts. Kane's dynamical equations have also been successfully used [10] in reducing the number of computations. Customized algorithms, and algorithms which ignore Coriolis and centrifugal forces in the dynamics formulation, have also been presented in order to reduce the number of computations [9], [10], [8]. Symbolic manipulation has also been attempted in an effort to increase the computational efficiency [1]. An extensive list of references on the computational aspects of robot dynamics can be found in [20].

With the advent of powerful CPUs, the dynamics computation can now be performed at relatively high speeds. Single CPU computers were found to be insufficient for some applications, however, and attempts were made to parallelize the dynamics problem. Towards this end, Luh and Lin [18] assigned one CPU to each manipulator link and proposed a variable branch-and-bound search algorithm to solve the dynamics of a Stanford arm. The computational speed of the same problem was improved using the depth first/implicit heuristic search algorithm [11]. On the other hand, Nigam and Lee [21] recognizing the recursive nature of dynamics suggested an algorithm which has a macro level pipeline, with parallelism within the elements of the pipeline. A parallel algorithm derived from the Newton-Euler state space model was discussed in Zheng and Chen [25] and an extension to that was presented by Zheng and Hemami [26]. Special VLSI implementable algorithms were also considered to reduce the computational speed [14], [15]. A completely parallel algorithm without any special scheduling was considered by Liu and Chen [16]. They proposed a scheme which utilizes one microprocessor for inverse kinematics and two for inverse dynamics and control, to solve the cartesian space control of robotic manipulators. Further work on multiprocessing of dynamics with a special scheduling scheme was considered by Vukobratovic et al. [23] and Chen et al. [2].

This paper addresses real-time implementation issues of dynamics simulations of robotic manipulators. This includes a discussion of the factors that contribute to the computational requirements, means of reducing computational loads and techniques for improving simulation execution time. A brief overview of dynamics simulation is also presented.

2 Dynamics Simulation

Dynamics simulation can be employed for the analysis and the design of robotic systems. Typical dynamic simulation activities include system dynamics response, analysis of vibrational modes of the structure, control system design, system verification, evaluation of kinematic design and operations support activities such as failure analysis. Dynamics simulations are often performed in real-time to support activities such as training and operations verification due to the fact that they provide a more realistic response than a kinematic simulation.

The dynamics of robotic manipulators deals with the relations between actuator torques and the link motions and includes the effects of the inertia of the links. Dynamics simulation is the process of determining the joint trajectories given the time histories of the joint torques, as well as the initial state of the manipulator.

In order to carry out a dynamics simulation, the equations of motion of the mechanical system must be formulated. The equations of motion can be represented in the following compact form:

$$M\ddot{q} = f + \tau \tag{1}$$

where M is the generalized mass matrix, q is the generalized coordinate vector, f is the generalized system force vector and τ is the external force vector. Equation 1 is arrived at by assembling the unconstrained equations of motions for each link and applying some transformation T in order to eliminate the constraint forces due to kinematic coupling [5] as follows:

$$M = T^T M_\Sigma T \tag{2}$$

$$f = T^T f_\Sigma \tag{3}$$

$$\tau = T^T \tau_\Sigma \tag{4}$$

where M_Σ is a constant unconstrained mass matrix, and f_Σ and τ_Σ are unconstrained system and external forces.

Equation 1 is then solved to obtain the vector of generalized accelerations $\ddot{q}$ for the system. These accelerations are then integrated in order to obtain the joint trajectories. Since M is symmetric and positive definite, Cholesky decomposition can be applied to facilitate rapid solution.

The formulation of the dynamical equations of motion can include rigid or flexible models of the joints and links that comprise the robotic manipulator. A rigid joint is one which produces rotational or translational displacement without any slippage and/or restraint. A flexible joint model may include joint deformation, stiction and backlash. Flexible links are typically modelled using shape functions.

3 Real-Time Dynamics Simulation Issues

3.1 Numerical Computation

Dynamics analysis through simulation requires the formulation of the dynamical equations, solving them to obtain the system accelerations and numerically integrating the accelerations to obtain velocities and displacements. All the three activities listed above can be computationally intensive. From equation 2, the computations needed to update the equations of motion increase with the cube of the number of simulated degrees of freedom [6]. From equation 1, the computations needed to solve the equations of motion increase with the square of the number of simulated degrees of freedom [6].

It has been found that for complex simulations [3], the majority of the total CPU time needed to update the equations of motion is taken by the transformation depicted by equation 2. Finding the eigen solution of the equations of motion, which is needed for model order reduction, can also be time consuming.

Stability considerations of the dynamics system can impose a requirement to perform the simulation at high iteration rates. Numerical integration methods of the variable time step and predictor-corrector type, such as the Gear's method, cannot be used because they do not provide deterministic execution times. Single time step methods must be used which require a very high iteration rate for their stability. Usually, the step size requirement of these methods are 10 times the highest frequency included in the dynamics model. In some applications, the iteration requirement may be influenced by other factors such as control system stablity or external interfaces to the simulator.

The iteration requirement together with the computation requirement can make real-time dynamic simulation with the current technology a formidable task. In order to simulate complex dynamic systems in real-time, specialized hardware and code optimization may be required. Various other techniques can also be employed to reduce computational loads.

3.2 Equations of Motion Update

Updating the equations of motion is the most computationally intensive task required in a dynamics simulation. Unacceptable delays in the outputs of a real-time simulation can arise if the solution of the equations of motion must wait for the coefficient updates. Moreover, updating the equations of motion each iteration may prevent the simulation from executing in real-time. Real-time requirements may force the equations of motion to be updated at a slower rate and force coefficient updates to be performed at the same time as the solution of the equations motion.

The computation of the generalized mass matrix and the generalized system force at slower rates will represent a considerable reduction in computational load but should not jeopardize the accuracy of the simulator. This is possible due to the fact that the coefficients of the equations of motion will often change at a slower rate than the state variables. The optimal rate of update can be found empirically by modifying the update rate and looking for convergence of results. Figure 1 compares simulations of a flexible manipulator with a 1 msec integration step size for various update rates of the equations of motion. The results begin to diverge significantly from the ideal result (i.e. 1 msec update period) when the update period exceeds 2 seconds.

If the coefficient and generalized coordinates updates are to be performed simultaneously, they must run as seperate processes. The processes can execute on seperate CPUs or if a single CPU is used, the coefficient update process must have a lower priority. Additional time will be required to synchronize data transfers between the processes.

3.3 Model Order Reduction

In the case of relatively slow moving robotic manipulators, it can be assumed that high frequency modes of the structure will not be excited. One of the techniques to eliminate high frequency components from the equations of motion is to use model order reduction. Because iteration rates depend upon the highest retained system frequency, model order reduction will relax the iteration rate requirements.

Model order reduction can be performed by removing high frequency modes from each body in the simulation. This technique, however, neglects effects that high frequency body modes may have on the overall system. A better approach is to find the eigen solution of the generalized equations of motion and then use the eigen vectors to transform the equations of motion to a modal space. The high frequency modes are removed by omitting the eigen vectors corresponding to high frequency modes from the modal transformation.

A simple method of performing model order reduction is to remove all modes above a given cut-off frequency. A more elaborate approach is to derive a cost function for each mode that is representative of some simulation output such as end-effector motion and remove modes that have no significant influence on the results. This approach is normally too expensive to implement for real-time simulations but can be useful for non real-time simulation in order to determine an adequate cut-off frequency for a real-time simulation.

Figure 2 demonstrates the effect of removing high frequency modes from the simulation of a flexible manipulator. It can be seen that keeping modes above 16 Hz has little effect on the simulation while removing modes above 3 Hz causes significant changes in the results.

3.4 Code Optimization

Most computers make use of high speed cache memory that can be accessed in a single instruction cycle. The CPU will look in the cache during code or data read operations. If the information is not in the cache it is read from the main memory which will introduce delays in the read operation. When a word of data is read from memory, the computer will typically load an entire block of memory (one cache line) into the cache. Minimizing the number of cache misses can represent a substantial portion of the code optimization process.

The total cache size and the cache line size will affect the number of cache misses in a given program. Complex simulations should derive greater benefit from a computer with a large cache. The overhead associated with cache misses depends upon the delay in loading a cache line. By pipelining memory operations, the effective loading time can be 1 cycle per word.

Dynamics simulation code involves many matrix operations that can benefit from cache optimization. Matrix operations should be stride minimized so that consecutive reads are performed on contiguous memory locations. This will minimize the number of cache misses while reading the data. For example, in the case that the compiler stores matrices in column major format, a routine which performs a matrix transpose times vector multiply will execute in less time than a regular matrix vector multiply. This is because there will be fewer cache misses when a matrix column is read than when a matrix row is read due to the difference in the stride. The arrays in a program should also be aligned with cache boundaries in order to make efficient use of the cache.

The optimization features of a compiler will attempt to take advantage of the architecture of the computer by performing parallelization, vectorization, pipeling and memory access optimization of operations. Due to the fact that the compiler may not always make the most efficient use of the CPU, hand tunning may be necessary. It may also be possible to use floating point library routines in order to make the maximum possible advantage of the computer system's architecture. Although highly efficent code can be produced by programming in assembly language, it should be avoided because of development costs and portability considerations.

3.5 Use of Specialized Hardware

Computing platforms that make use of specialized hardware and software in order to execute more than one instruction at a time can be employed to reduce execution times. Such platforms may support vectorization and/or multiple CPUs sharing common memory.

The algorithms for updating and solving the equations of motion are each generally recursive in nature and cannot be easily split up to run on multiple CPUs. In some cases it may be beneficial to parallelize or vectorize certain operations such as matrix multiplication, especially when the number of degrees of freedom is large.

Joint level control system code may also lend itself to parallelization if the degrees of freedom are decoupled. A computing platform that can perform vector operations in H/W may also provide significant performance improvement for large numbers of degrees of freedom.

When a simulation task is split among multiple CPUs, there will be overhead due to the operating system, increased bus traffic and synchronization of data exchanges. Variations in execution times will be typically more pronounced than on a single CPU machine due to memory contention between processors. It has been seen that for elastic multibodies [4], lack of process synchronization can cause instability in the simulation. Failure to properly synchronize data transfers will render a simulator that, at best, cannot produce deterministic results. Moreover, the time required for development and integration of simulation software on a multi-CPU platform will be greater due to the added effort involved in synchronizing data transfers between processes running on seperate CPUs. Simulation code written for a single CPU will have an advantage over code written for multiple CPUs because it will be less hardware dependant and hence easier to port to new platforms. Portability is an extremely important issue in real-time simulation due to the rapid evolution of computer technology.

Platforms that support hardware vectorization will require additional time associated with the loading of vector registers and the extraction of results. Experience has shown that this additional time makes such platforms ineffective when compared with CPUs using RISC methodology for many dynamics simulations.

Dynamics simulation code is dominated by floating point operations. The most common operations performed in dynamics simulation are matrix-matrix and matrix-vector multiplies which can be decomposed into vector products. A vector product requires the CPU to load two floating point registers, multiply them together and then add the result to a third register. A CPU that provides an effective throughput of a few cycles by pipelining the operation, would be ideal for dynamic simulation. This approach has been adopted by many of the latest CPUs using RISC methodology.

Floating point registers must be loaded from cache in order to provide high throughput due to the speed at which regular memory operates. A program will not benefit from cache memory if the data is used only once before being flushed out. This is often the case in dynamics simulations where operations are performed on large matrices and the data in the cache is replaced before the matrix is required again. A computer that supports the loading of floating point registers directly from main memory at the same effective rate as loading the cache from main memory should be capable of improving dynamics simulation execution times.

4 Conclusions

Real-time dynamic simulation is a computationally demanding activity even for simple 6 link manipulators due to the complexity of the equations of motion and because of stability considerations that force the simulator to run at high iteration rates. The coefficient update normally must be performed in parallel with the solution of the equations of motion to avoid delays in the outputs. This should not present a problem since the coefficients change more slowly than the state for most systems. Model order reduction can be used to remove high frequency modes from the system making the simulation more stable. This helps to relax the constraints that are placed on the integration step size.

Multiple CPUs and/or specialized vectorizing CPUs can be used to reduce execution times provided that the benefit due to parallelization out weighs the additional execution time and development costs. However, a single CPU machine using RISC methodology to execute floating point instructions with few clock cycles will provide better performance than a multiple CPU machine using standard floating point hardware for most applications. Moreover, a simulator written for a single CPU platform will be much easier to port as computer technology evolves.

5 Acknowledgements

We would like to thank CAE Electronics Ltd. for permitting this article to be presented.

References

[1] Cetinkunt, S., Siciliano, B., and Book, W.J., 1986, "Symbolic modeling and dynamic analysis of flexible manipulators," *Proc IEEE Int. Conf. on Syst. Man. and Cybern.*, pp. 798-803.

[2] Chen, C. L., Lee, C. S. G., and Hou, E. S. H., 1988, "Efficient scheduling algorithms for robot inverse dynamics computation on a multiprocessor system," *IEEE Int. Conf. on Robotics and Automation*, pp. 1146-1151.

[3] Crabtree, D., Buhariwala, K., D'Eleuterio, G. D., and Scott, P., 1993 "Real time simulation of multi-body dynamics - application to the space station freedom mobile servicing system," *44th Congress of the International Astronautics Federation*

[4] Cyril, X., 1991, "MDSF Validation Report: RT Dynamics Simulation(V1.0)", *CAE Technical Document*, Rev. 1.1.

[5] Cyril, X., 1991, "Dynamic analysis of complex space structures using natural orthogonal complement," *42nd Congress of the International Astronautics Federation*

[6] Cyril, X., Angeles, J., Misra, A., 1991, "Dynamics of flexible multibody mechanical systems," *Transactions of the CSME* Vol. 15, No. 3. pp 235-256.

[7] Hollerbach, J. M., 1980, "A recursive Lagrangian formulation of manipulator dynamics and a comparitive study of dynamic formulation complexity," *IEEE Trans. Syst., Man Cybern.*, Vol. SMC-10, No. 11, pp. 730-736.

[8] Horak, D. T., 1984, "A simplified modeling and computational scheme for manipulator dynamics." *ASME J. Dynamical Systems, Measurement, and Control*, Vol. 106, pp. 350-353.

[9] Horn, B. K. P. and Raibert, M. H., 1978, "Configuration space control," *The Industrial Robot*, pp. 69-73.

[10] Kane, T. R., and Levinson, D. A., 1983, "Use of Kane's dynamical equations in robotics," *Int. J. Robotics Res.*, Vol. 2, No. 3, pp. 3-20.

[11] Kasahara, H., and Narita, S., 1985, "Parallel processing of robot-arm control computation on a multiprocessor system," *IEEE J. Robotics and Automation*, Vol. RA-1, No. 2, pp. 104-113.

[12] Kazerounian, K., and Gupta, K. C., 1986, "Manipulator dynamics using the extended zero reference position description," *IEEE Trans. Robotics ans Automation*, Vol. RA-2 No. 4, pp. 221-224.

[13] Khalil, W., Kleinfinger, J. F., and Gautier, M., 1986, "Reducing the computational burden of the dynamic models of robots," *Proc. IEEE Int. Conf. on Robotics and Automations*, pp. 525-531.

[14] Lathrop, R. H., 1985, "Parallelism in manipulator dynamics," *Int. J. Robotics Research*, Vol. 4, No. 2, pp. 80-102.

[15] Lee, C. S. G., and Chang, P. R., 1986, "Efficient parallel algorithm for robot inverse dynamics computation," *IEEE Int. Conf. Robotics Automation*, pp. 851-857.

[16] Liu, C. H., and Chen, Y. M., 1986, "Multiprocessor based cartesian space control techniques for a mechanical manipulator," *IEEE Int. Conf. Robotics Automation*, pp. 823-827.

[17] Luh, J. Y. S., Walker, M. W., and Paul, R. P. C., 1980, "On-line computational scheme for mechanical manipulators," *ASME J. Dynamical Systems, Measurement, and Control*, Vol 102, pp. 69-76.

[18] Luh, J. Y. S., and Lin, C. S., 1982, "Scheduling of parallelcomputation for a computer-controlled mechanical manipulator," *IEEE Trans. Syst., Man Cybern.*, Vol. SMC-12, No. 2, pp. 214-234.

[19] McInnis, B. C., and Liu, C. K. F., 1986, "Kinematics and dynamics in robotics: A tutorial based upon classical concepts of vectorial mechanics," *IEEE Trans. Robotics and Automation*, Vol. RA-2, No. 4, pp. 181-187.

[20] Newman, C. P., and Murray, J. J., 1985, "Computational robot dynamics: Foundations and applications," *IEEE J. Robotics Automation*, Vol. RA-1, No. 4, pp 173-182.

[21] Nigam, R., and Lee, C. S. G., 1985, "A multiprocessor based controller for control of mechanical manipulators," *IEEE J. Robotics Automation*, Vol. RA-1, No. 4, pp. 173-182.

[22] Silver, M. W., 1982, "On the equivalence of Lagrangian and Newton-Euler dynamics for manipulators," *Int. J. Robotics Res.*, Vol. 2, No. 3, pp. 60-70.

[23] Vukobratovic, M., Kircanski, N., and Li, S. G., 1988, "An approach to parallel processing of dynamic robot models," *Int. J. Robotics Research*, pp. 64-71.

[24] Walker, M. W., and Orin, D. E., 1982, "Efficient dynamic computer simulation of robotic mechanisms," *ASME J. Dynamic Systems, Measurement, and Control*, Vol. 104, pp. 205-211.

[25] Zheng, Y. F., and Chen, B. R., 1985, "A multiprocessor for dynamic control of multilink systems," *Proc. IEEE Int. Conf. Robotics Automation*, pp. 295-300.

[26] Zheng, Y. F., and Hemami, H., 1986, "Computation of multibody system dynamics by a multiprocessor scheme," *IEEE Tran. Syst., Man Cybern,.* Vol. SMC-16, No. 1, pp. 102-110.

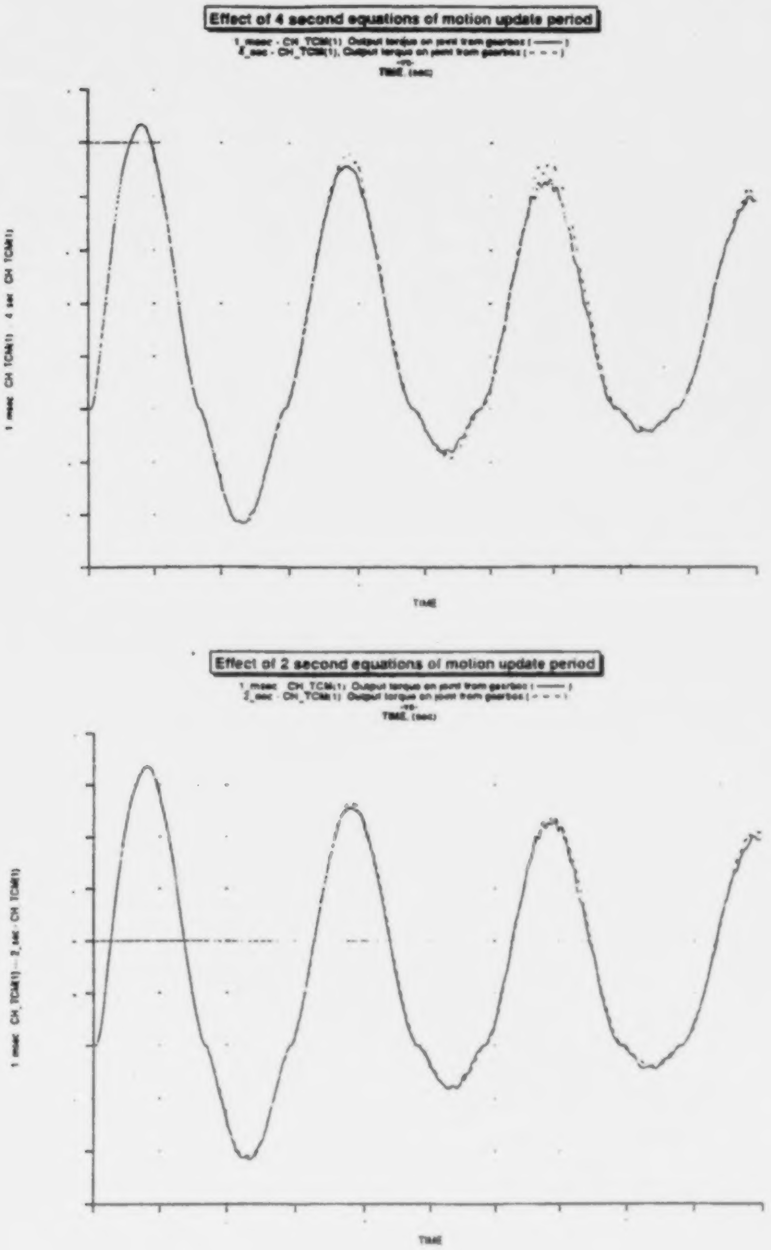


Figure 1: Effect of Equations of Motion Update Rate on a Flexible Manipulator

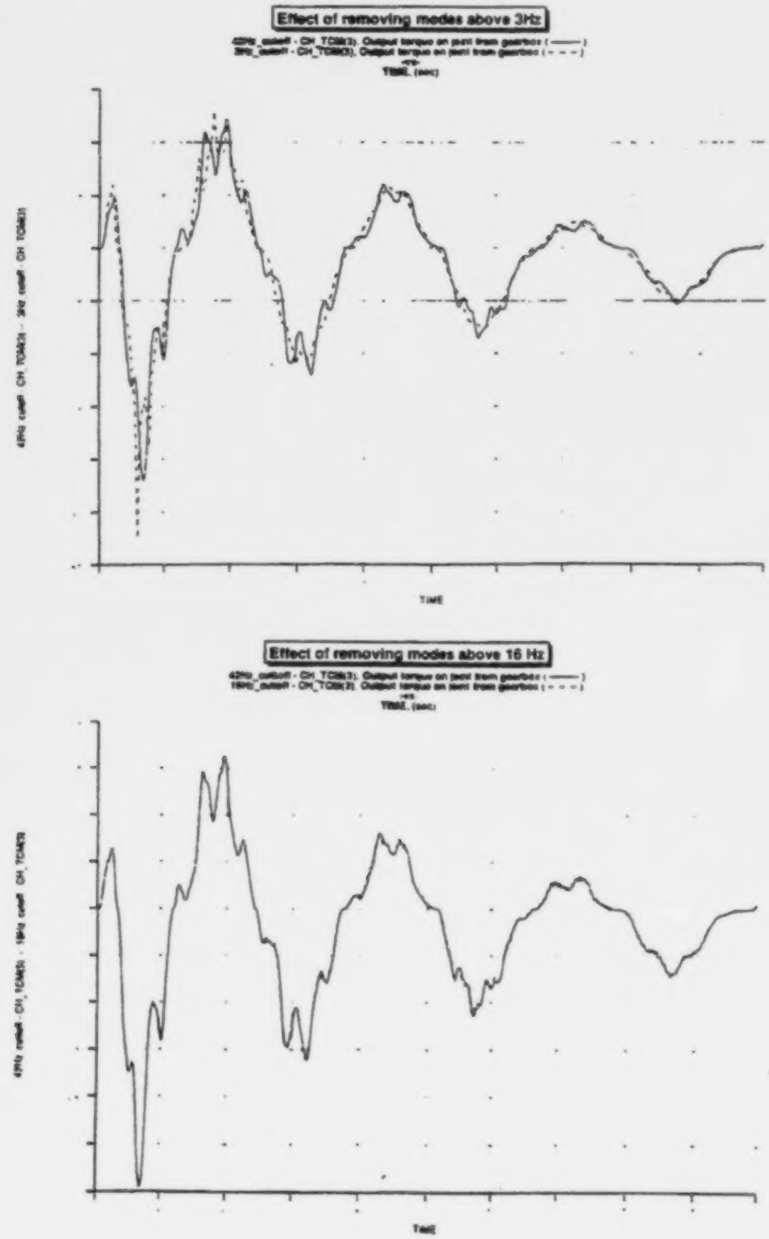


Figure 2: Effect of Model Order Reduction on a Flexible Manipulator

90C208

Robotic Interface for CAD/CAM Integration

Martin Ferland
Hydro-Québec
5100, rue Sherbrooke Est
Montréal (Québec) Canada H1V 3R9

Jules O'Shea and Monjy Rabemanantsoa
École Polytechnique
Case postale 6079, succursale A
Montréal (Québec) Canada H3C 3A7

149724

Abstract

The problem of convertibility of CAD simulation data into CAM activities in the fields of robotic applications is discussed.

CATIA (Computer Aided-graphic Three Dimensional Interactive Application) is adopted as the basis for building robot structure because of its capabilities of defining the joints, motions and tasks. Thus, a specific method has been designed for translating and converting the simulation output file into VAL language. The significance of this method comes from the new approach of converting the robot data record by means of an interface called "CONVERT" written in C language. The "CONVERT" interface has been applied to several simulations involving a mechanical assembly sequences and "Towers of Hanoï " sequences. Sequences match PUMA robot motion and operation in the real world.

Keywords: *CAD/CAM, Robotic, Task Simulation, VAL/Interface, Program converter.*

Introduction

For many years, some predefined tasks in the robotics field have been solved using sensor-based reasoning and/or trajectory generation algorithms in conjunction with off-line programming. Generally, these procedures use constraints of the system to implement operation sequences for achieving specified goals. The basis for establishing a set of design rules for robotic and automated assembly are presented with some generalized design guidelines by Jackson and McMaster [1]. Sanderson & al.[2] introduced a method for the task planning of robotic manipulation in space applications. Comments are made about autonomous systems incorporating embedded task planning with accompanying sensory integration, control synthesis, and system architecture to support goal-directed activities in an uncertain environment.

Based on this reasoning, a specific method of converting CATIA simulation output to VAL is presented to drive the PUMA robot. Also, robot solid modelling and task simulation associated with high-level design are addressed. The simulation generates a file named CATASK which provides information related to robot model, geometric entities, motion description and task parameters. An important use of these database facilities is the construction of an intelligent interface to bridge the gap between the CAD environment and CAM. The interface called CONVERT provides a framework for handling and integrating data along with a mechanism for managing data, robot programs and the manipulation of the data with programs.

Conceptual Design of the Integration

A new approach is determined for studying task planning from the point of view of robotic simulation of CATIA/CAD system. Our methodology breaks the steps into three areas:

First, all the modelling considerations are centered around a feature-based design scheme. The object definition based on set primitives like cuboid, cylinder, cone, etc., provides a 3D-object representation of solids. Complex shapes can be designed by means of "constructions and combinations" using set operators: union, intersection and difference. One of the best features of solid modelling is the elimination of redundancy in the representation during the design process.

Second, the method describes an approach which uses computer-based reasoning on geometry about the description of joints and kinematics. A constraint-based geometric reasoner has been developed as part of a robot motion system. The robot model is accessed through its features. The overall implementation has been done in conjunction with the PATH option to define different mode for achieving the study of curve-following trajectory, collision avoidance and trajectory generation.

The third contribution is dedicated to the interface development. The output record contains data defining the task simulation; the CONVERT interface interacts with that record for integrating both geometric and technical information of robot and environment. A data translation method provides a knowledge communication mechanism from the CAD design to the robotic task-planning.

Finally, the major contribution of the authors' approach is in the knowledge representation of constraint-based geometry related to joints and kinematics and the translation and conversion of task file into VAL statements. The objective is to supply a "pattern accessible structure" to capture design considerations and knowledge for integrated manufacturing. In the following discussion we present the research methodology with related aspects describing the steps towards CAD/CAM integration feasibility.

Robot Solid Modelling

CAD representation has increased over the past few years. "Solid Modelling" methodology has been adopted for a complete and consistent information about parts [5]. Robot modelling uses the SOLID function of CATIA to create and process volume and surface ELEMENTS. All ELEMENTS are created in the current SET. During the various construction phases of robot structure, canonical volumes are generated from the CREATE menu:

- CREATE: CUBOID, PIPE, PRISM, PYRAMID, TRANSFORM, REVOLUTION, CYLINDER, CONE, TORUS, SPHERE, UNSPECIFIED.

These volumes may be combined to gradually generate more complex form. "Constructions and Combinations" are achieved by use of mathematical operators called "Boolean" into the OPERATION menu:

- OPERATION: UNION, INTERSEC, SUBTRACT, SPLIT, ASSEMBLE.

Figure 1 illustrates solids used to model the PUMA robot (dimensions are in centimeters).

Description of Joints and Kinematic

Robot modelling is completed by adding joints. Joints and kinematics are described into CAD/system for purpose of studying the motions and the static loads exerted on the mechanism. Three types of joints exist: Revolute which are controllable by angular command, Prismatic which are controllable by linear commands and Coupling Revolute which are not controllable by commands. The coupling revolute definition allows keying two angles limiting the travel in relation to the current configurations and the limited positions are displayed. Joint creation is generally followed by a three-axis system attached to the

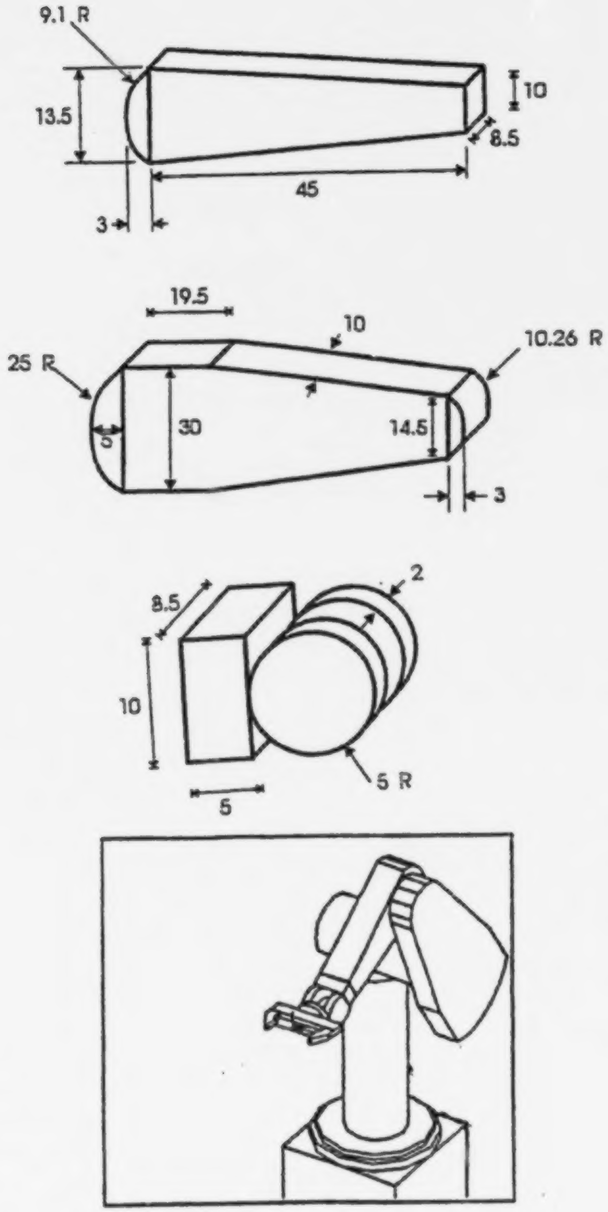


Figure 1: Puma Robot Solid Modelling

terminal parts to identify its position. In this context, a fixed reference axis related to the base of the robot is first defined.

The kinematic parameters of the robot consist of velocities and possibly accelerations/decelerations or upstroke and downstroke times. These features are taken into account during motions, namely, the velocities of the end effector during linear interpolation trajectory, and the joint velocities during joint motion using joint interpolation. It is very convenient to use the modes (i.e. joint mode or linear mode) available in CATIA to tell the robot system what is the desired goal point of the manipulator motion is:

- in joint mode, the display steps are associated with the revolute and prismatic joints
- in linear mode, the display steps are associated with the orientation and translation of the 3-axis reference system associated to the effector.

Figure 2 shows how the joint motions of the robot are presented on the screen; the arrows must be interpreted in the sense of the rule of thumb.

Task - Planning Simulation

The TASK level is achieved within the frame work of the user's conceptual model of the process. It deals with a sequence of primitive actions, i.e., place, rotate, insert where each action must be mapped into a sequence of these operations. Therefore, the TASK function has been addressed to off-line programming and within CATIA this function is used to simulate sensors and use it in a job cycle.

The PATH option is used for all motions to define interpolation modes, accelerations and decelerations and curve following tolerances. The interpolation mode consists either in "joint interpolation" which synchronizes the moving parts of a joint or in "linear interpolation" which makes sure the selected elements of the

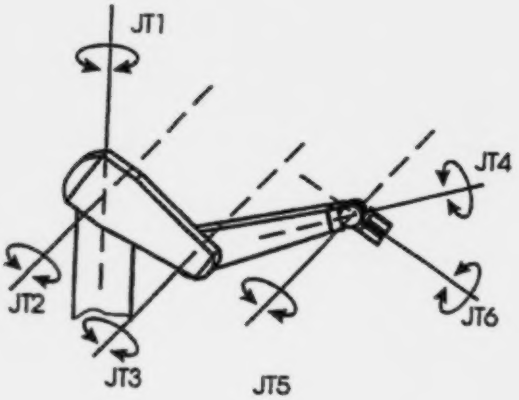


Figure 2: Joint Motions of the Robot

robot move in a straight line. The SPEED command is used to set the acceleration and deceleration values. The underlying three-dimensional data base associated with robot motion is very convenient. In the case of a curve mapped onto a surface, orientation constraints are simply defined with respect to the normal direction and tangent for curve following trajectory. For every elementary operation (grasp, release, ...), we associate an "attribute" (for example force exerted for grasping an object) and this latter figures out in the output file for programming the real robot.

During task execution any mismatch between task parameters and robot capabilities results in an error diagnosis. It is activated either if the cartesian position to be reached is not part of the robot's work space or if the robot cannot reach the expected joint position. The "mode window" is used to manage the definition mode of task.

Simulation goes beyond kinematic model and integrates sensors for the purpose of interference checking and trajectory generation. In addition, setting the WORKCELL function enables several tasks to be executed and synchronized. An execution of a process may then be evaluated for optimization. A "window mode" of the system controls the definition mode followed by presenting modifiable parameters such as: COLS (collision monitoring), TRACE (geometric elements trace option); at level execution, we deal with CONT (continuous task mode) or BY_STEP (by working in "discrete" mode where the task can be verified by being executed instruction by instruction).

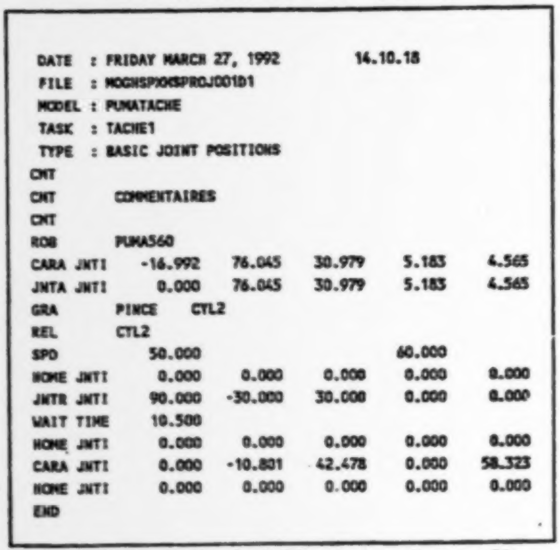
File Translation and Conversion

When the simulation ends and the "sequences" has been adjusted, the CATASK file is used to write all the characteristics of the task to be executed and to convert them to VAL statements. CATASK is generated by the CAD system. The record contains a data defining task in a file whose type is named FICTASK. All the numerical values are written in F10.3 format (10 characters with 3 decimals). Identifiers and strings are recorded under 8 characters. The "basic" output file describes commands and motions along with the robot joint positions or the cartesian positions of its reference three-axis systems. The "advanced" output file positions are registered either in relative values or absolute values. This file takes into account the task logical structure and describes variables and all the instructions.

The purpose of the "CONVERT" interface, is to deal with the CATASK file. It translates and converts the simulation information into VAL program to drive the robot. Although the set of data is very large, a data structure has been developed for analyzing the task and optimizing the various instructions under the most favourable conditions. Pointers are associated to I/O (input/output) data fields and a global variable is used to key in the identifier of existing joints in order to translate their coordinates and values.

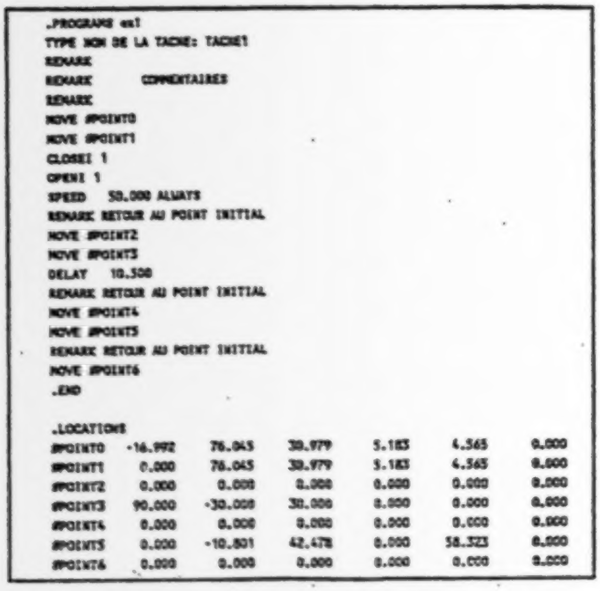
In the conversion phase, considerations are mainly centered around motion parameters.

"CONVERT" provides a scheme to store and manipulate these features using the standardized set of operators. Figure 3 shows a file generated by the CAD system; figure 4 illustrates the same file after conversion.



```
 DATE  : FRIDAY MARCH 27, 1992          14.10.18
 FILE  : MOGHSPIOKSPROJ001D1
 MODEL : PUMATACHE
 TASK  : TACHE1
 TYPE  : BASIC JOINT POSITIONS
CNT
CNT       COMMENTAIRES
CNT
ROB       PUMA560
CARA JNTI    -16.992    76.045    30.979     5.183     4.565
JNTA JNTI      0.000    76.045    30.979     5.183     4.565
GRA       PINCE   CTL2
REL       CTL2
SPD          50.000                         60.000
HOME JNTI     0.000     0.000     0.000     0.000     0.000
JNTR JNTI    90.000   -30.000    30.000     0.000     0.000
WAIT TIME    10.500
HOME JNTI     0.000     0.000     0.000     0.000     0.000
CARA JNTI     0.000   -10.801    42.478     0.000    58.323
HOME JNTI     0.000     0.000     0.000     0.000     0.000
END
```

Figure 3: Typical Simulation Output file



```
.PROGRAME ex1
TYPE NOM DE LA TACHE: TACHE1
REMARK
REMARK       COMMENTAIRES
REMARK
MOVE #POINT0
MOVE #POINT1
CLOSEI 1
OPENI 1
SPEED   50.000 ALWAYS
REMARK RETOUR AU POINT INITIAL
MOVE #POINT2
MOVE #POINT3
DELAY   10.500
REMARK RETOUR AU POINT INITIAL
MOVE #POINT4
MOVE #POINT5
REMARK RETOUR AU POINT INITIAL
MOVE #POINT6
.END

.LOCATIONS
#POINT0   -16.992    76.045    30.979     5.183     4.565     0.000
#POINT1     0.000    76.045    30.979     5.183     4.565     0.000
#POINT2     0.000     0.000     0.000     0.000     0.000     0.000
#POINT3    90.000   -30.000    30.000     0.000     0.000     0.000
#POINT4     0.000     0.000     0.000     0.000     0.000     0.000
#POINT5     0.000   -10.801    42.478     0.000    58.323     0.000
#POINT6     0.000     0.000     0.000     0.000     0.000     0.000
```

Figure 4: Output File after Conversion

Conclusion

The objective of this approach is to supply a new integrated interface in the sense of CIM (Computer Integrated Manufacturing). The knowledge-state of the manipulator motions and operations are defined and validated before accessing the real robot. Such test simulations are helpful for the applications in the fields of nuclear reactors, space telerobotics and automated assembly systems.

The design methodology of the research has two related aspects. At the CAD level, a real representation of the robot in 3D space via solid modelling is used to define and study the motions taking into account the degrees of freedom. Robot kinematics are also defined in terms of end effector velocities and joint velocities. At the CAM level, the interface that supports a multiple-information environment provides a specialized form of interprocess communication.

The "CONVERT" interface is built around the concept of an integrated scheme in factory automation. The information management system supports both data definition and data manipulation. Thus, it provides interesting model-stored processes for job cycle and task-planning optimization.

References

1. A. J. Jackson, R. S. McMaster, "Product Design for Robotic and Automated Assembly", *Proc. of Int. Conf. on Robotics and Automation,* Vol. 2, IEEE, 1986.

2. A. C. Sanderson, M. A. Peshkin, L. S. Homem de Mello, "Task Planning for Robotic Manipulation in Space Applications", *IEEE Trans. on Aerospace and Electronic Systems,* Vol. 24, No 5, Sept. 1988, pp. 619-628.

3. F. Pfeiffer and R. Johanni, "A Concept for Manipulator Trajectory Planning", *IEEE Journal of Robotics and Automation,* RA-3, 1987, pp. 115-123.

4. R. A. Jarvis, "Configuration Space Collision-Free Path Planning for Robotic Manipulators", *Robots in Australia's Future Conference* , ARA, 1988, pp. 194-204.

5. Y. C. Lee and K. S. Fu, " A CSG Based DBMS for CAD/CAM and its Supporting Query Language", *CAD Journal, CH 1886-1/83/0000/0123,* IEEE, 1986, pp. 123-128.

6. IBM, "CATIA Robotics User's Manual", *Dassault System,* 1991.

7. Unimation Robotics, User's Guide to VAL (398P2A), 500 Series, *Unimate Puma Mark II Robot,* Vol. 2, Feb. 1983.

8. R. F. Woodbury and I. J. Oppenheim, "An Approach to Geometric Reasoning in Robotics", *IEEE Trans. on Aerospace and Electronic System,* Vol.24, No.5, Sept.1988, pp. 630-645.

9. B. O. Nnaji, "A Framework for CAD-Based Geometric Reasoning for Robot Assembly Language", *Int. Journal Proc. Res.,* «vol.26, No.5, 1988, pp. 735-764.

10. P. H. Gu, H.A. ElMaraghy, "FDDL: A Feature-Based Design Description Language", *Journal of the ASME,* 1990, pp. 53-63.

11. M. Rabemanantsoa, S.Pierre, "A Knowledge-Based Approach for Robot Assembly Planner", *Proc. of the Canadian Conf. on Electrical & Computer Engineering,* Vancouver, Canada, IEEE Vol.2, Sept.1993, pp. 829-832.

90534C

Application of Object Oriented Analysis and Design Methodology to Knowledge Based Systems Development - The Cable Run Intelligent Advisor System

Stephen Jones[1], Constantinos Bassias[2], Raymond Sassine[3] & C. Tousignant[4]
Paramax Systems Canada Inc.,
6111 Royalmount Avenue,
Montreal, Quebec, Canada H4P 1K6

#149727

Abstract

Use of Object Oriented (OO) methods and programming techniques helps in modeling the domain, identifying formalism of the knowledge, building applications very fast, and reusing the general structures whenever appropriate. This paper discusses the approach associated with the transformation of a real world problem into a suitable information model using object-oriented knowledge representation. The main advantages of applying the object-oriented paradigm are that, at the analysis phase, it introduces a software development guided framework for performing knowledge engineering while, at the design stage, it provides a more direct modeling flow from software requirements to design elements.

The Object Oriented paradigm was used at Paramax in the development and implementation of a knowledge based application, the Cable Run Intelligent Advisor (CRIA) system, which is an engineering tool that facilitates the design, configuration and configuration management tasks of the cable run engineering process as well as the fabrication of cables in large complex electronic systems. This process is an intricate one involving an extensive set of specific knowledge elements which spans from the basic engineering guidelines, standards, methods and techniques to those of data and configuration management. From a user specification the system produces a cable assembly and/or wire harness which conforms to standards and constraints, and minimizes the overall design cost.

The analysis approach used on this project was adapted from the Object Oriented Analysis (OOA) methodology described by Yourdon and Coad [Yourdon E., Coad P., "Object Oriented Analysis", Prentice-Hall, 1990.]. This allowed a representation of the domain into objects and classes which, in conjunction with conventional slots and rules, provides a suitable method of knowledge representation.

The Cable Run Problem

The Cable Run Engineering Process consists of design, production specification, configuration management and fabrication materials management of cable assemblies and/or wire harnesses. This is a challenging problem, particularly when applied to large complex platforms like the Canadian Patrol Frigate (CPF) and the New Shipborne Aircraft (NSA).

There are over 4,000 cables on the CPF. Each cable is associated with two connections; each connection is associated with at least two and up to more than one hundred individual pin-outs, each in turn associated with particular wires within the cable, each of which carries specific signal data or power. The Cable Run design experts must research the data from a wide variety of platform and/or equipment specific documents and drawings, and must apply hundreds of individual facts and rules, to properly design and specify the instructions by which the cables are manufactured.

---

[1] Phone (514) 340-8447
[2] Phone (514) 340-8674
[3] Phone (514) 340-8708
[4] Phone (514) 340-8333

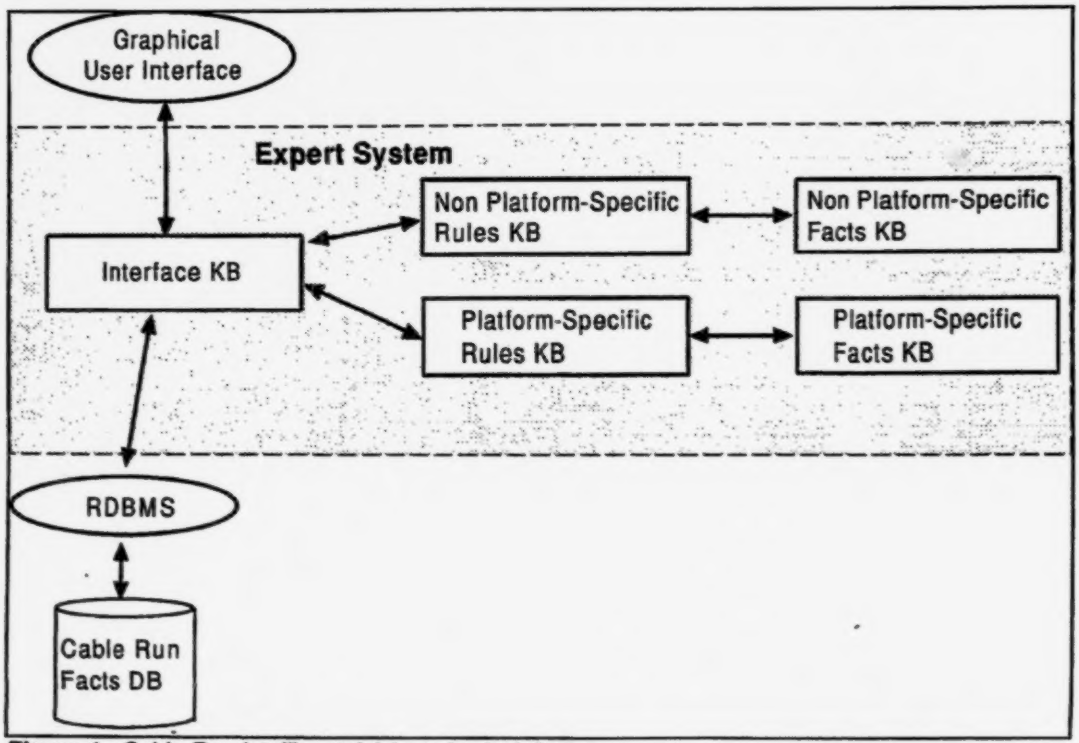

Graphical User Interface
Expert System
Non Platform-Specific Rules KB
Non Platform-Specific Facts KB
Interface KB
Platform-Specific Rules KB
Platform-Specific Facts KB
RDBMS
Cable Run Facts DB

Figure 1 *- Cable Run Intelligent Advisor physical design*

The Cable Run Intelligent Advisor System Concept

The objective of the Cable Run Intelligent Advisor System (CRIA) is to help automate the processes by which specialized engineering experts design and configuration manage the electrical wiring and cables on large-scale programs.

Figure 1 shows that the Cable Run Intelligent Advisor System is composed of three main modules: a knowledge-base (expert system) component, a user interface component, and a database component.

The knowledge-base component instantiates the knowledge about the design, change management, and reporting standards of cable run or wire harness configuration.

The user interface passes the user's commands through to the appropriate internal processes, and relays back the appropriate results.

The actual Cable Run data is stored and managed through the Relational Database Management System (RDBMS) component.

a) User Interface Component

The User Interface (UI) comprises two parts: Interface KBs and an X-Window Graphical User Interface (GUI) program. This front end interface transparently implements all of the necessary calls to the expert system and database components. The UI component allows the user to enter new information, or to retrieve and update existing information, minimizing entry errors through extensive data validation. Its main components are: a configuration interface knowledge base, which triggers the knowledge processing for the configuration inside the expert system component, an information interface knowledge base, for updating and querying the database, and an X-Window program containing the routines for displaying the knowledge and information being processed by the CRIA system. Additional components include a reporting facility, several utility functions and a direct path through Structured Query Language (SQL) to the INGRES database.

b) Database Component

This component is a relational database divided into more than 100 tables which instantiate the relational model of the platform-specific and platform-independent paradigm used in the overall design. This component also includes specialized reporting and querying procedures which are used in methods in the knowledge bases.

Applying The Object Oriented Paradigm

The Object Oriented Analysis (OOA) approach used on this project was adapted from the methodology described by [Yourdon 90]. This approach allowed a representation of real-world entities such as equipments, their integral connection points, their pin-outs, etc.., into a corresponding set of object classes and nonplatform-specific knowledge bases. Figure 2 shows an example of the equivalence between a real-world equipment, at one end of a Cable Run, and the corresponding CRIA objects.

CRIA Objects can be thought of as simultaneously active little machines, that each do a job. While the piece-parts (sub-objects) of each object may remain visible in a local state, and while various objects may have conditional interdependencies in certain processes, their ability to participate in a particular process is arbitrated by the CRIA expert system, on the basis of rules derived from our OO analysis, of their services and their attribute properties.

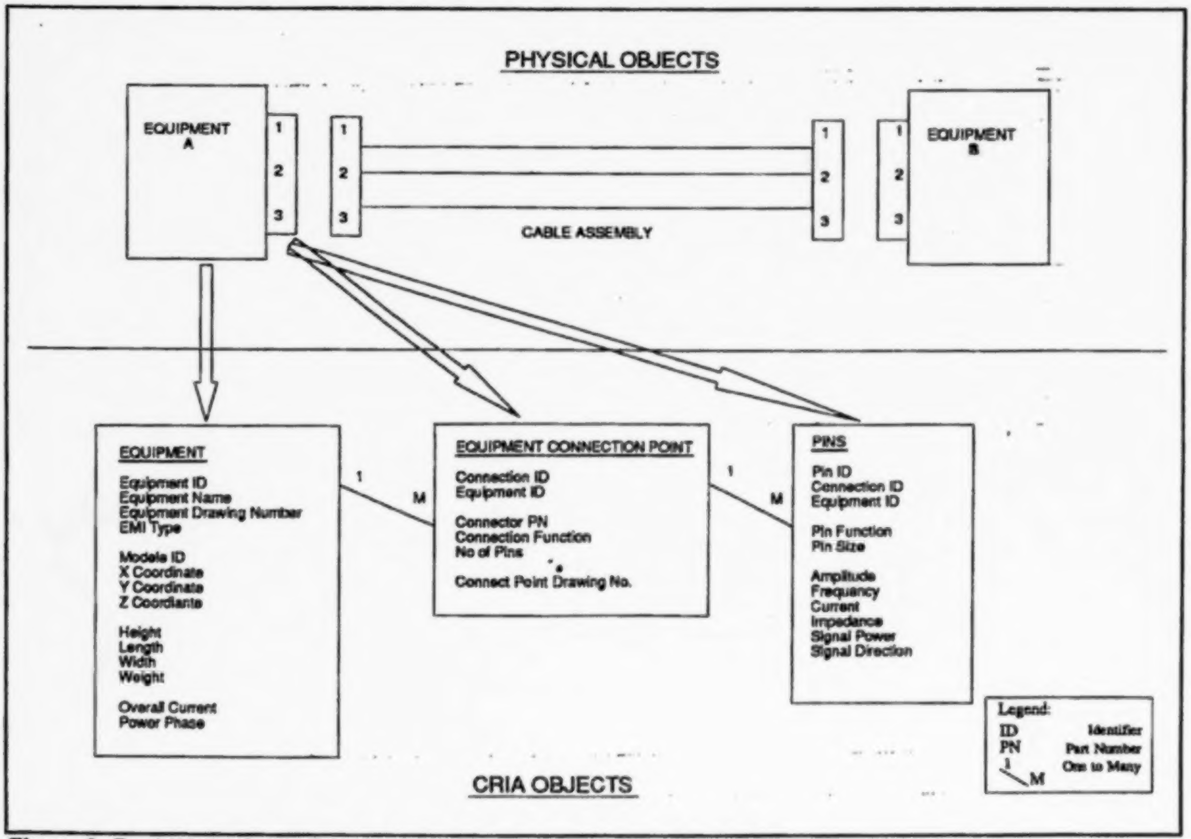

PHYSICAL OBJECTS
EQUIPMENT A
1
2
3
1
2
3
CABLE ASSEMBLY
1
2
3
1
2
3
EQUIPMENT B
EQUIPMENT
Equipment ID
Equipment Name
Equipment Drawing Number
EMI Type
Modele ID
X Coordinate
Y Coordinate
Z Coordiante
Height
Length
Width
Weight
Overall Current
Power Phase
1
M
EQUIPMENT CONNECTION POINT
Connection ID
Equipment ID
Connector PN
Connection Function
No of Pins
Connect Point Drawing No.
1
M
PINS
Pin ID
Connection ID
Equipment ID
Pin Function
Pin Size
Amplitude
Frequency
Current
Impedance
Signal Power
Signal Direction
Legend:
ID Identifier
PN Part Number
1 M One to Many
CRIA OBJECTS

Figure 2: Real World Cable Run Objects Map Directly into CRIA Objects

Modeling Knowledge and The OO Paradigm

Choosing a knowledge representation scheme is often the most critical decision a knowledge-based system developer makes when applying software engineering technology to a design problem. This particular decision is critical because the organization of knowledge makes certain manipulations of data easy and others difficult or impossible.

The choice of knowledge representation schemes is usually influenced strongly by the class or nature of the design problem (e.g., planning problem, classification problem, design problem, etc.). This is appropriate since classes of problems tend to place certain requirements on the knowledge representation schemes. Thus a key factor in efficient design system is the classification and the representation of the types of design knowledge needed.

In an object oriented environment, there are two conventional forms of knowledge representation, declarative and procedural. The two forms are applicable to different categories of knowledge. For example, declarative knowledge describes an object itself in terms of its properties. Procedural knowledge, on the other hand, describes how an object relates to its environment. The latter form of knowledge usually consists of methodologies. Both of these representations are needed to implement a knowledge-based design tool.

The separation between declarative and procedural knowledge is made explicit in the Cable Run Intelligent Advisor.

A) Classification of Knowledge

The design objects, in the case of cable assemblies and wiring harnesses, are very complex; from a structural view they are composed of many parts, each of which possesses different dimensions and properties. For example the description of a Cable Run Sheet includes many components; the header which contains the specific characteristics of the cable (length, type...), and which describes the systems involved in the connection; the detailed wiring list, which describes the characteristics of the connection pin to pin; and a last component which defines the additional requirements to complete the cable assembly.

Any attempt at representing both the objects and the design expertise is likely to be similarly complex. Typically, Cable Run expert design knowledge can be divided into three categories:

i) Structural knowledge about the connectors, cables and wires, e.g., the selection of the type of materials and pin sizes, etc.

ii) Factual knowledge relating to cables/wires dimensions, parameters and performance.

iii) The designer's expertise which is used to provide directional control for the design process. This type of knowledge is the most abstract and tends to be the most difficult to acquire and represent.

The CRIA attempts to incorporate the kind of knowledge used by a designer in optimizing cable assemblies and/or wire harnesses design. Much of this knowledge is non-exact and provides short cuts through the complete life cycle and field analysis of the design.

B) Object Oriented Knowledge Representation

When one attempts to incorporate structural knowledge of a problem into the knowledge-base, an OO construct provides a suitable mechanism for describing the structure of a part. Inferencing can then occur based on structural relationships within the problem domain. For example, in the case of connectors-pin outs, the slots describe each of the input/output pins of the main components, and in turn, point to further objects. Each component is described by its physical properties and function characteristics. Each of these objects provides a level of abstraction which reduces the complexity of the search in the database.

C) Rule-based Knowledge Representation

In a rule-based system, all knowledge of the domain is expressed in the form of rules. Production rules generally adhere to a modular and consistent structure; each rule is often expressed in the form of situation-action pairs, i.e., they express what to do if a particular situation occurs. In the cable run design system, several different families of rules are needed for both controlling the design and for determining the values of the properties of each object. Provided the rule base is complete and correct, (i.e., no two rules conflict in their actions in a particular situation), then an application of a search strategy based on the rules should achieve the required solution.

D) Knowledge-bases for CRIA

The knowledge-bases (KBs) of the CRIA system contain all of the knowledge relating to cable design and reporting which must be applied to the automated processes. For the CRIA to perform as efficiently as possible, it was necessary to determine what types of knowledge were to be represented, and to group the related rules together with each other but separately from other independant rules. OOA representation in conjunction with conventional slots and rules abundantly described in the literature [Buchanan 85][Waterman 86][Prerau 90], also provided a suitable method of knowledge representation.

Our analysis of the Cable Run design processes lead us to conclude that the CRIA knowledge-base should consists of three distinct modules: the Configuration knowledge module; the Reporting knowledge module; and the Change management knowledge module.

i) The configuration knowledge base module contains the rules that determine a valid configuration design and components for the connection of two specified equipments.

This knowledge base is used to support users in the definition of cable configurations.

ii) The Reporting Knowledge base module contains all the rules for determining the actual report information required for final output.

This knowledge base ensures that the Cable Run Sheet deliverables are complete and compliant with the contractual and engineering requirements.

iii) The Change management knowledge base module keeps track of approved engineering change requests and their associated contractual documentation.

This knowledge base ensures that configuration management of the cabling data is maintained in accordance with the relevant program and process requirements.

Porting the CPF CRIA Application to a Second Platform (NSA)

The close correlation between real-world objects and CRIA objects resulting from our application of the OOA approach allowed us to reuse 80% of the CPF object structure in the NSA application.

The major difference between the CPF and the NSA requirements was that aircraft are cabled with individual wires, as opposed to the sealed cable assemblies used on ships. (There were other functional enhancements requested by the NSA program team, but only the wiring strategy difference is relavant to this discussion.)

In order to analyse the adaptation of CRIA to NSA, the same OOA approach was applied:

- a review of the Classes and Objects identified the need to revise the objects and rules associated with cables into new objects and rules defining wires and their management;
- a comparison of the structures showed that the same equipment-related data structures apply, the database simply requires populating with the NSA platform-specific information; and
- a review of the standards and specifications governing the EMI/EMC design considerations will result in revisions to those platform-specific rules.

The NSA CRIA design specifications have been completed in 40% of the time required for the original CPF CRIA system. The savings are directly attributable to the successful application of Object Oriented Analysis and Design strategies, in the development of the original system.

Conclusions

In developing the CRIA, some advantages and some disadvantages of applying an OO methodology for the development of a knowledge-based system were found.

The primary advantages were:

a) OO proved to be a convenient method by which to develop a general purpose class system to design the domain-specific CRIA/CPF and CRIA/NSA objects equivalent to the real-world facts and materials;

b) as a result of the object-oriented design approach, the porting of the CRIA application from a naval platform (CPF) to an air platform (NSA) was also quite straight-forward; and

c) the Facts Knowledge Bases produced can be easily modified and enhanced when necessary. This can be done without affecting their integration into the overall system.

The major disadvantage was that the rule base of the CRIA system could not be designed with the Object Oriented methodology. There was no categorical analysis of Knowledge Elements, thus the CRIA rule base is not actually structured except for a split between platform dependant and platform independant rules. (The Common KADS (Knowledge Acquisition Design System) methodology would be a more appropriate approach to rules base design than OOA/OOD.)

References

Buchanan B., Shortliffe E.,[1985] "*Rule Based Expert Systems*", Addison Wesley Publishing Co., 1985.

Waterman D.A., [1986] "*A Guide to Expert Systems*", Addison Wesley Publishing Co., 1986.

Prerau. D.R., [1990] "*Developing and Managing Expert Systems*", Addison Wesley Publishing Co., 1990.

Yourdon E., Coad P., [1990] "*Object Oriented Analysis*", Prentice-Hall, 1990.

90C207

CLAP: Une implantation du modèle Acteur sur réseau hétérogène

Jocelyn Desbiens, Michel Toulouse et Denis Gagné
Centre de Recherche en Informatique Distribuée (CRID)
Département d'informatique et d'ingénierie CYC
Collège militaire royal de Saint-Jean
Richelain (Québec) Canada
J0J 1R0
Adresse électronique: `{desbiens,toulouse,dgagne}@cmr.ca`

#149729

Résumé

Le modèle Acteur de Agha et Hewit [1, 12, 13] est une extension naturelle au domaine de la programmation parallèle du paradigme d'objet tel que nous le connaissons. Le modèle respecte les principes premiers de la programmation objet: abstraction des données et encapsulation, héritage ou délégation, communication par envoi de messages, modularité et homogénéité. Cependant, il y ajoute celui d'activité: chaque acteur est un agent actif, autonome, communiquant librement avec ses semblables. Nous décrivons dans le présent article une implantation du modèle Acteur sur réseau hétérogène: le langage d'acteurs CLAP. L'article se scinde en trois parties: 1) nous rappelons les grandes lignes de la définition du modèle Acteur, 2) nous discutons de l'implantation, présentement en phase finale, du modèle Acteur sur machine à mémoire distribuée ainsi que sur réseau de stations de travail et 3) nous donnons des détails sur le fonctionnement d'une application multi-agents et du mécanisme de son immersion dans le système CLAP.

Introduction

Le but de la présente communication est de présenter un langage d'acteurs implanté sur réseau hétérogène constitué de stations Unix et de machines parallèles à mémoire distribuée: le système CLAP (C++ Libraries for Actor Programming) ainsi qu'une application en cours de réalisation dans cet environnement. Cette application, le système Aurora, consiste à simuler à l'aide d'un système multi-agents l'ensemble des interactions produites par un équipage d'avion militaire lors d'une mission.

Le système CLAP a été développé en premier lieu pour programmer les ordinateurs parallèles à mémoire distribuée. Il fut par la suite porté sur réseaux de stations de travail Unix. Le modèle de calcul réalisé par CLAP est basé sur le paradigme Acteur. L'implantation de CLAP a été réalisée en C++ et la bibliothèque de classes l'accompagnant lui rajoute les notions de communication asynchrone et de concurrence en plus des principaux concepts du modèle Acteur. Les principales classes de cette bibliothèque sont: Actor, Behavior, Message et Scheduler. CLAP est un environnement de programmation permettant le développement et la gestion des applications en intelligence artificielle distribuée. Le niveau d'abstraction sur lequel est basée CLAP permet de faire cohabiter et fonctionner des applications hétérogènes dans un milieu virtuellement homogène.

Le système Aurora est un simulateur d'équipage d'avion militaire basé sur le concept de multi-expertise. Un système multi-experts est une "équipe" d'agents autonomes travaillant en commun, suivant des modes de coopération pour aboutir à un objectif global: la résolution d'un problème. Chaque expert, fonctionnant sur son processeur indépendamment des autres experts, représente un des membres de l'équipage de l'avion. Il possède sa propre base de faits ainsi que sa propre vision du monde et communique avec les autres experts via l'envoi de messages. Tous les experts travaillent de concert à la réalisation d'un but commun: trouver et détruire le ou les sous-marins. Aurora a été développé indépendamment du système CLAP et l'immersion dans le système s'est faite en quelques jours de travail, démontrant du même coup la souplesse et la polyvalence de l'approche Acteur pour la programmation des applications distribuées.

Dans ce travail, nous donnerons en premier lieu un bref aperçu du modèle Acteur [1, 12, 13] et ensuite une description du système CLAP, une réalisation du modèle Acteur sur réseau hétérogène. Finalement, nous parlerons du projet Aurora et nous donnerons, dans la dernière section, quelques détails du mécanisme d'immersion et de développement du système multi-agents Aurora à l'intérieur de l'environnement CLAP.

Langages à base de classes

Un langage de programmation accepte une notion de type abstrait s'il permet à l'usager de définir des sous-ensembles de structures de données et d'opérateurs qui ont accès exclusif à ces structures. La programmation par objets naît d'un prolongement de cette notion de type abstrait. Pour beaucoup de langages à objets, la notion de type abstrait est réalisée sous forme de modules par la définition de protocoles d'interface. On obtient de cette façon la propriété d'encapsulation des données qui est grosso modo une sorte d'effet de bord de la notion de type abstrait, puisque l'accès aux données entre modules est régi par le protocole d'interface. Seuls les opérateurs définis à l'intérieur du module accèdent directement à ces données.

Certaines contraintes inhérentes au développement de logiciels de grande taille ont fait que la notion d'encapsulation est devenue un sujet d'expérimentation pour le développement de nouveaux langages de programmation. Au plan des langages à objets, l'encapsulation des données par les modules fait en sorte que ceux-ci possèdent leur propre état (ou mémoire interne) qui est mis à jour uniquement par les opérateurs du module. On parle alors d'objets pour désigner ces modules. Les modules objets ont eu un impact certain sur la façon de modéliser la résolution de problèmes. Le traitement est en effet plus fortement décomposé, il est réalisé par un ensemble de petits programmes relativement autonomes collaborant entre eux par échanges de messages pour aboutir à la solution finale du problème. La programmation d'une application consiste par conséquent à définir les objets composant le problème et le type de protocole d'interface de chaque objet.

Les modules objets combinés avec la notion d'encapsulation ont permis de simplifier le développement de gros logiciels. Une des promesses de l'émergence de la notion d'objet en génie logiciel fut la "réutilisation du code". En principe, n'importe quel objet peut envoyer un message à un autre objet pour lui demander de faire un calcul particulier, évitant ainsi d'avoir à réécrire le code qui effectuait ce calcul. On voyait déjà l'existence de bibliothèques d'objets généraux et on pensait que le développement d'un logiciel consisterait à utiliser ces bibliothèques génériques de pair avec les objets plus spécifiques des applications. De cette préoccupation pour la réutilisation du code à travers la spécialisation des objets est née la notion de niveaux d'objets, plus particulièrement la notion d'héritage.

Certains langages du paradigme objet combinent étroitement cette notion d'héritage avec une structure hiérarchique de types. Les objets sont des instanciations d'une structure (la classe, le type) décrivant le format et le comportement de l'objet. Les objets n'héritent plus directement les uns des autres. Ce sont les classes qui sont le lieu de l'héritage du format et du comportement des objets donnant ainsi naissance à une hiérarchie de types. Les classes introduisent deux nouvelles catégories de relations entre les objets d'un système: la relation *is a* qui indique qu'un objet est une instance d'une classe et la relation *kind of* qui indique que la classe d'un objet est une sous-classe de la classe d'un autre objet. Certains langages offrent des primitives permettant d'exprimer directement ces nouvelles relations entre les objets. Smalltalk en est un exemple. On fait référence à ces langages sous l'appellation de langages à base de classes. Malheureusement, dans la plupart de ces langages, la communication entre un objet et sa classe ou entre une classe et sa sous-classe ne se fait pas par envoi de messages. Dans ce cas, la notion d'héritage entre directement en conflit avec celle d'encapsulation et, règle générale, le principe d'encapsulation est affaibli.

Le calcul parallèle

Les modèles de calcul parallèle

Il existe plusieurs modèles du calcul parallèle. Sans doute l'un des plus connu et aussi l'un des plus conservateur est le modèle CSP de Hoare [15]. Un programme dans ce modèle est constitué d'un nombre fixe de processus séquentiels communiquant uniquement par envois de messages. Cette communication entre les processus séquentiels est synchrone et la topologie des processus est déterminée de manière statique. La communication synchrone rend donc impossible l'écriture de programmes récursifs dans le modèle de Hoare: un processus qui est occupé (*busy wait*) à expédier un message ne peut recevoir de messages. Le fait que la topologie des processus soit statique rend inutile la reconfiguration dynamique qui pourrait être offerte par certaines architectures de machines parallèles.

Un autre modèle plus radical est celui de la programmation fonctionnelle de Henderson [10]. Le parallélisme dans la programmation fonctionnelle est le résultat du fait que tous les arguments d'une fonction peuvent être évalués concurremment. Cependant, un problème capital avec la programmation fonctionnelle est que les fonctions n'ont pas d'historique. On peut prendre l'exemple bien connu du compte de banque. Le comportement d'un compte de banque change avec le temps en fonction de la balance du montant dans le compte. Or une fonction ne se rappelle pas l'état d'une précédante activation (en programmation fonctionnelle pure, ce n'est pas le cas cependant en LISP). Certaines solutions ont été proposées pour ce problème de la programmation fonctionnelle mais elles ne sont guère satisfaisantes [2].

Le troisième est celui du modèle Acteur. Le modèle Acteur fut développé au MIT par Hewit [11, 12, 13] et son équipe pour exploiter l'envoi de messages comme base du calcul parallèle. Essentiellement, un acteur est un agent de calcul autonome qui exécute ses actions en réponse à la réception d'un message. Les acteurs sont des modèles locaux de calcul et il n'existe pas d'état global de tous les acteurs dans le système.

Le modèle de calcul Acteur

La notion d'encapsulation est strictement respectée dans les langages du modèle Acteur et dans ce sens on fait référence à ces langages comme étant un sous-ensemble des langages du paradigme objet. Dans le modèle Acteur, l'acteur joue un rôle analogue à celui de l'objet dans les langages à objets. Un acteur est défini par son *script*, c'est-à-dire ce qui correspond à la définition d'un module objet dans les langages à objets. Un acteur peut être vu comme:

- une adresse, à laquelle se trouve une boîte aux lettres contenant les messages reçus par l'acteur,
- un comportement, qui correspond aux actions entreprises par l'acteur pour traiter le message courant.

Les acteurs interagissent sur une base purement locale en s'expédiant des messages [13]. La sémantique des actions décrites par un script est très simple. Un acteur peut:

- envoyer un message (*send*),
- créer un ou d'autres acteurs (*new*),
- spécifier sa mutation (*become*).

Un acteur est connu dans le réseau des acteurs d'un système par son adresse, c'est ce qui permet à un acteur de changer de comportement tout en gardant la même identité. À des fins de modélisation, on peut faire correspondre un comportement à un *script*: un acteur ayant plusieurs comportements dans le temps possède donc plusieurs *scripts*. Cependant, à un instant donné, un acteur se comporte selon la description d'un seul *script*.

Le terme "acteur" lui-même met l'accent sur un autre aspect de la programmation dans le modèle Acteur, c'est-à-dire la programmation par processus concurrents. Le paradigme objet général a été conçu pour un environnement monoprocesseur où les objets sont séquencés (*scheduled*) à tour de rôle. Par opposition, un "acteur" désigne non seulement le code du programme de l'utilisateur mais aussi l'environnement d'exécution de ce code. Dans ce sens, un acteur est actif contrairement à un objet qui ne possède pas sa propre pile d'exécution.

La notion de classe n'existe pas dans le modèle Acteur. La création d'un nouvel objet (acteur) se fait par clonage et non par instanciations d'un type comme dans les langages à bases de classes. L'objet où est prise la copie du clone se nomme prototype. Le nouvel objet peut devenir à son tour un prototype pour un autre objet. Par omission ou par ajout de propriétés, la nouvelle copie peut ne pas être identique à son prototype. C'est ainsi que de façon incrémentielle il est possible de modéliser un problème dans un langage Acteur sans avoir à faire usage d'une hiérarchie de classes. Plusieurs chercheurs [5, 19, 21, 22, 24] n'ont pas manqué de souligner ce fait.

La notion d'héritage est également absente du modèle Acteur original. La seule façon dont deux objets peuvent s'échanger de l'information est de passer par le protocole d'interface des modules. C'est cette propriété importante du modèle Acteur qui garantit que le principe d'encapsulation sera respecté. Du point de vue de la programmation parallèle, cette propriété redonne au paradigme objet tout son intérêt parce qu'alors les langages Acteur ont des propriétés forçant l'exécution des programme à être semblables à la sémantique de leur *script*.

Le modèle original de Hewit [12] de la programmation Acteur ne prévoyait pas de mécanisme de partage de code. C'est Lieberman [19] qui le premier en 1986 proposa un mécanisme de réutilisation du code dans le modèle Acteur qui ne violait pas le principe d'encapsulation, c'est-à-dire le mécanisme de la délégation.

La délégation est l'approche au partage des connaissances dans le modèle Acteur. C'est un ensemble de conventions portant sur l'envoi de messages entre acteurs pour obtenir le même effet que l'héritage en Smalltalk. L'idée fondamentale de la délégation est que lorsqu'un acteur ne sait pas comment répondre à une requête, il doit la communiquer à un ou plusieurs acteurs qu'il connaît afin de transmettre le résultat à l'acteur source de la requête. Ces acteurs à qui un acteur donné peut déléguer les requêtes sont identifiés comme étant les *proxies* de cet acteur. Un *proxy* joue un rôle analogue à celui d'une super-classe dans le schéma d'héritage.

Le protocole de communication du schéma de délégation est uniforme: la délégation au *proxy* d'un objet se fait par envoi de message, et non pas par un système de pointeur. Ce schéma est donc indépendant *a priori* de l'architecture de la machine (mémoire partagée ou distribuée).

Le langage CLAP

Le développement à partir de zéro d'un nouvel environnement de programmation est une aventure généralement longue, coûteuse et risquée. C'est la raison pour laquelle nous avons choisi de restreindre de prime abord l'effort de développement strictement aux nouveaux concepts qui interviennent dans l'environnement de programmation que nous visions. Les mécanismes qui acceptent les hiérarchies de classes des langages objets rendent possible la création d'abstractions pour différents modèles de programmation sans qu'il soit nécessaire de faire des modifications à la définition du langage de base. Smalltalk [4], C++ [23] et ABCL/1 [25] ont été quelques uns des langages de base que nous avons étudiés en vue de servir de plate-forme de développement de CLAP. Nous avons choisi C++ , *primo*, parce qu'il s'agit d'un langage compilé et donc *a priori* plus performant que Smalltalk ou ABCL/1, *secundo*, parce qu'il existe des compilateurs croisés C++ générant du code pour les plates-formes matérielles visées et, *tertio*, parce que nous avons pu obtenir la bibliothèque de classes du système ACT++, un système développé par Kafura [17] de Virginia Tech. University et dont les objectifs se rapprochent passablement des nôtres.

Comme nous l'avons vu ci-dessus, Acteur est un modèle de programmation concurrente où les objets sont actifs et communiquent entre eux de manière asynchrone. Or le modèle de calcul que présuppose C++ est synchrone et séquentiel, c'est-à-dire qu'il n'existe à chaque instant qu'une seule trame de contrôle (*thread*) exécutant un programme C++. Si une méthode est invoquée en C++, son exécution est lancée immédiatement en utilisant la trame de contrôle qui exécutait la méthode appelante. La première étape, dans la fusion du modèle Acteur et de C++, fut d'écrire un certain nombre de classes rajoutant à C++ ces notions de concurrence et de communication asynchrone. Les concepteurs de ACT++ ont résolu ce problème en utilisant PRESTO, une bibliothèque de classes en C++ développée à l'Université de Washington. Il était impossible d'utiliser PRESTO. En effet, la plate-forme parallèle ciblée possède une architecture à mémoire distribuée alors que PRESTO présuppose l'existence d'une architecture à mémoire partagée. La bibliothèque de classes de CLAP comprend donc des classes qui font appel à des primitives du système d'exploitation Helios en vue de réaliser la communication asynchrone et l'activation d'objets avec piles d'exécutions indépendantes. Sur les stations de travail Unix, nous avons utilisé les "*lightweight process*".

La deuxième étape de la combinaison du modèle Acteur avec C++ fut d'écrire les classes implantant les abstractions du modèle Acteur. Ces abstractions ne différant guère d'un langage Acteur à un autre, il a été possible d'utiliser les classes existantes de la bibliothèque de ACT++. Il a été virtuellement impossible d'utiliser les méthodes de ces classes sans les modifier. Cette première version de CLAP implante les concepts suivants du modèle Acteur: la notion d'acteur, les comportements, la boîte à lettres et le parallélisme au niveau des modules. Le tout sur plusieurs processeurs dans un environnement distribué. Ce qui n'a pas été implémenté dans cette version sont les mécanismes de délégation et de clonage. Nous avons ajouté le parallélisme intra-acteur au modèle Acteur original. Ce nouveau niveau de parallélisme permet de désérialiser jusqu'à un certain point le traitement des messages. Les classes les plus importantes qui implantent le modèle Acteur dans CLAP sont Actor, Behavior, Message et Scheduler. Nous allons traiter en quelques mots des classes Actor, Behavior et Scheduler dans cette section.

La classe Actor

Chaque acteur créé par le programme d'un usager est une instanciation de la classe Actor de la bibliothèque du système CLAP. La création d'un acteur dans l'environnement de programmation distribué CLAP doit se faire différemment que dans ACT++. Nous avons ajouté la méthode CreateActor à la classe Actor de ACT++. Le programme de l'usager envoie un message à CreateActor chaque fois qu'il veut créer un nouvel acteur. Un des arguments de la méthode CreateActor est le nom du nouvel acteur utilisé pour fin d'enregistrement dans le système. Cette méthode retourne au créateur de l'acteur un numéro identifiant de façon unique le nouvel acteur dans le système. Il est possible d'avoir accès à chaque acteur soit par son numéro, soit par son nom.

La classe Behavior

La classe Behavior détermine les actions qu'un acteur peut prendre lorsqu'il traite un message. Dans le modèle Acteur, un acteur est un objet ayant plusieurs protocoles d'interface, c'est donc dire qu'un acteur peut se comporter de plusieurs façons différentes. Pour chaque comportement d'acteur, le programmeur doit définir une sous-classe de la classe existante Behavior. Chacune de ses sous-classes contient les structures de données ainsi que la définition des méthodes qui peuvent manipuler ces structures de données.

La principale méthode de la classe Behavior est *become* dont le rôle est de changer le protocole d'interface de l'acteur après ou pendant le traitement de chaque message. Comme nous l'avons vu dans la section précédente, le protocole d'interface est le moyen par lequel les objets peuvent traiter les messages qu'ils reçoivent. La notion de changement de protocole d'interface au cours de l'existence d'un même acteur dans le modèle Acteur joue deux rôles principaux:

- il permet aux acteurs d'avoir un comportement distinct et leurs donne accès à l'historique du calcul,
- il permet à un acteur de traiter plusieurs messages à la fois, ajoutant ainsi un autre niveau de concurrence à celui de l'exécution concurrente de plusieurs acteurs dans le système.

Le *become* peut cependant ne pas être localisé à la fin du *script* d'un acteur et permettre ainsi un niveau de parallélisme intra-acteur. Le *become* ainsi que le traitement du prochain message peuvent être entamés, même si le traitement du présent message n'est pas terminé. On peut de cette façon avoir plusieurs copies du même acteur exécutant en parallèle, la fonction du *become* devenant celle d'un sémaphore garantissant l'accès exclusif par un processus aux variables partagées de l'acteur dans la section du *script* précédant le *become*. C'est la responsabilité du programmeur de s'assurer qu'il est placé au bon endroit afin que les variables partagées par d'éventuels processus concurrents soient mises à jour en respectant la sémantique du programme.

La classe Scheduler

Comme plusieurs acteurs cohabitent sur un même processeur, il est nécessaire d'avoir un mécanisme robuste de changement de contexte. La classe Scheduler dans CLAP gère l'utilisation des ressources d'un processeur par les acteurs. Le nombre de méthodes dans Scheduler a considérablement été réduit par rapport à la classe correspondante de ACT++. Son utilité dans la présente version de CLAP se limite, sur les machines parallèles à mémoires distribuées, à lancer l'exécution d'un acteur pour le traitement d'un message. Par la suite, c'est le séquenceur micro-codé du processeur T800 qui gère le temps d'exécution des processus. Ce séquenceur micro-codé est évidemment très performant mais, d'un autre côté, trop rudimentaire. Ainsi,

- il ne permet pas d'avoir plus de deux niveaux de priorité,
- il est sans préemption pour les processus de plus haut niveau de priorité,
- il séquence les processus uniquement pour certaines instructions assembleur qui ne sont pas nécessairement engendrées par le compilateur C.

Il faudra probablement ajouter à la classe Scheduler certaines méthodes afin de remédier aux problèmes du séquenceur micro-codé.

La programmation en CLAP

Quand il écrit un programme CLAP, le programmeur utilise la même syntaxe que celle de C++ et doit connaître l'existence des méthodes CreateActor de la classe Actor et *become* de la classe Behavior. Le programmeur détermine uniquement les différents acteurs en jeu dans son système. Tout le reste est laissé au système.

Un tel programme est d'abord passé au filtre du pré-processeur. Ensuite, les fichiers sources de l'usager sont compilés par un compilateur croisé générant du code C. Ce code C, avec les deux fichiers générés par le pré-processeur, est compilé et lié avec les bibliothèques de classes du système CLAP pour ensuite générer du code exécutable sur les T800 et les processeurs RISC. Au chargement du programme, le programmeur spécifie sur quel processeur sera chargé chaque classe de son programme. Le système CLAP est présent sur chaque transputer utilisé par le programme ainsi que sur chacune des stations de travail Unix faisant partie du réseau de calcul.

Une application CLAP

CLAP vu comme langage de simulation

Le paradigme de la programmation objet remonte à Simula [4], un langage de programmation dérivé d'Algol 60 [20]. Simula est un langage spécialisé pour traiter des problèmes de simulation. Les notions fondamentales de la programmation objet (objet, classe, programmation concurrente, ...) étaient déjà présentes dans le langage au tout début de sa création. En Simula, un objet est un programme actif autonome pouvant communiquer et se synchroniser avec d'autres objets. Ces qualités utiles à la simulation se sont propagées et il n'est donc pas étonnant que le langage CLAP, en tant que réalisation sur réseau hétérogène du paradigme d'Acteur et *a fortiori* de la programmation objet, puisse être vu comme un environnement naturel de modélisation et de simulation du monde physique.

La facilité avec laquelle il est possible de paralléliser les solutions d'un problème est fonction du domaine d'application de ce problème. Ainsi, une simulation se retranscrit bien dans un environnement Acteur car il existe dans ce cas une correspondance biunivoque naturelle entre, d'une part, les composantes du monde simulé et leurs communications et, d'autre part, le monde des acteurs et les messages que les acteurs s'échangent en vue de se synchroniser. En revanche, dans le cas d'une application où la décomposabilité du problème est moins évidente (par exemple, lorsque l'on veut immerger une application déjà existante dans un environnement objet), ce sont les structures de données du problème et les opérateurs agissant sur ces structures qui pourront être mis en relation avec les acteurs du système et leurs messages.

Le système Aurora

Un exemple de logiciel de simulation écrit en CLAP est le système Aurora: il s'agit de simuler l'ensemble des interactions produites par un équipage d'avion chasseur de sous-marins lors d'une mission. Le but ultime du projet est de fournir aux Forces Armées Canadiennes une plate-forme matérielle et logicielle d'entraînement d'équipages d'avions (Aurora ou autres) où une partie, sinon la totalité, des membres de l'équipage seraient remplacés par des agents synthétiques intelligents. À court terme, les objectifs du présent projet sont plus de démontrer la faisabilité d'une architecture multi-agents dans un environnement distribué que d'en arriver à un système opérationnel d'entraînement d'équipage d'avion militaire. Pour plus de détails sur le système Aurora, nous renvoyons le lecteur aux références [7, 8].

Cette simulation sur machine parallèle illustre bien la décomposabilité du domaine visé (simulation d'une société hiérarchisée d'agents interagissant entre eux de manière bien définie) et sa formulation en termes d'acteurs. Le modèle comprend plusieurs types d'acteurs:

- les objets dynamiques représentant les composantes premières de la simulation (avion chasseur de sous-marin Aurora, sous-marins, navires, ...). Chacun de ces acteurs comprend un ou plusieurs groupes d'acteurs représentant les membres de l'équipage ou les instruments qui leur sont affectés.
- les experts représentant les membres de l'équipage. Chaque système expert simule l'un des membres d'équipage d'un avion militaire de détection de sous-marins. Leurs fonctions vont du pilotage de l'appareil au contrôle tactique de la mission, en passant par la supervision des instruments de détection et de l'armement de l'appareil. Chacun des experts est un acteur "intelligent" en ce sens qu'il simule le mode de raisonnement d'un être humain et est programmé comme une instance d'un système expert. La coquille de systèmes experts est une implantation sur transputer et sur Sparc d'une coquille disponible commercialement [26, 27]. En plus d'un moteur d'inférence à chaînage avant, cette coquille développée par la NASA, possède plusieurs caractéristiques non standards (objets, débogueur intégré, etc.). Les coquilles CLIPS communiquent entre elles via le système CLAP par échange standard de messages entre acteurs. Par exemple,

```
(Pilote (nouveau_cap 450))
```

est un message envoyé par le navigateur au pilote de l'Aurora lui signifiant une nouvelle direction à prendre. Remarquons qu'il n'est pas nécessaire ici de spécifier l'adresse du pilote car le navigateur et le pilote font partie du même agglomérat d'acteurs: l'acteur Aurora.

- les instruments qui sont des acteurs plus simples réagissant aux requêtes des experts. Chacun de ces acteurs ne répond qu'aux messages adressés par l'expert auquel il est associé. Par exemple, seul le pilote de l'Aurora a la possibilité de faire changer de cap l'aéronef, tous les autres messages en provenance d'autres acteurs étant filtrés par CLAP.

Dans la simulation en CLAP du système Aurora, tout est acteur. Chacune des composantes est donc définie en termes de ses variables locales et des méthodes s'appliquant à ces données. Par exemple, l'acteur CLIPS (le "moteur" du raisonnement de chacun des experts) a la définition suivante:

```
class clips:public Behavior{
        int         filedes[2];

public:
        clips(char *);
        void    commande(chaîne);
        void    départ();
};
```

Tous les experts (en réalité des coquilles de système expert CLIPS augmentées de base de faits et de règles correspondant aux interactions générées par l'équipage de l'avion) sont des instanciations de la classe `clips`. Celle-ci est dérivée de la classe Behavior qui indique à CLAP que cette classe représentera un acteur au cours de l'exécution de la simulation. On remarque aussi la présence de deux méthodes appartenant à la classe, si l'on fait bien sûr abstraction du constructeur de la classe. Ces deux méthodes (`commande` et `départ`) sont les deux seuls messages que pourra recevoir ce type d'acteur en cours d'exécution. Le message `depart` permet de lancer un `clips`, par parallélisme intra-acteur, qui sera constamment à l'écoute d'un tube (*pipe*). Ce tube sera utilisé par le message `commande` afin de communiquer à ce `clips` les informations reçues des autres experts.

Conclusion

Il existe trois approches bien connues du développement des langages de programmation distribuée:

- la parallélisation automatique de code séquentiel se basant sur l'analyse des dépendances de données (PTRAN, Paraphrase II et PTOOL);
- ajout de primitives à des langages foncièrement séquentiel (ABCL/1, Ada, C parallèle et Prolog Concurrent);
- conception de langages nouveaux pour les architectures parallèles (SISAL, VAL, Linda et Occam).

Le système CLAP appartient à la troisième catégorie. Dans ce système nous avons essayé de faire converger les caractéristiques de la programmation objet et la résolution de problèmes en programmation distribuée. Une première étape a donc consisté, dans une version préliminaire du système, à implémenter la majorité des concepts du modèle Acteur. Cette filiation avec le modèle Acteur a permis de développer un système de programmation par objets préservant les caractéristiques intéressantes du paradigme objet dans un environnement physique de machines parallèles et stations de travail Unix.

Un domaine d'application important de l'informatique est celui de la simulation. Le système Aurora est un modèle de simulation qui a été réalisé en CLAP et qui illustre bien la relation étroite existant entre les structures fondamentales de la simulation et les concepts essentiels de la programmation objet.

Références

1. G. Agha, « ACTORS: A Model of Concurrent Computation in Distributed Systems », The MIT Press, Cambridge, Massachusetts, 1986.
2. G. Agha, « Foundational Issues in Concurrent Programming », *SIGPLAN Notices*, vol. 24, n° 4, pages 60-65.
3. J. Bloomer, « Power Programming with RPC », O'Reilly & Associates, Inc eds., 1991.
4. G. Birtwistle, O.J. Dahl, B. Myhrhaug et K. Nygaard, « Simula begin », Petrocelli Charter, New York, 1973.
5. A.H. Borning, « Classes versus prototypes in object-oriented languages », *Proceedings of the ACM/IEEE Fall Joint Computer Conference*, 1986, pages 36-40.
6. D. Coromel, « A generalized model for concurrent and distributed object-oriented programming », *Proceedings of the ACM SIGPLAN Workshop on Object-Based Concurrent Programming*, 1988.
7. J. Desbiens, M. Janta-Polczynski et G. Nault,« Projet Aurora: Systèmes Experts parallèles

5. A.H. Borning, « Classes versus prototypes in object-oriented languages », *Proceedings of the ACM/IEEE Fall Joint Computer Conference*, 1986, pages 36-40.

6. D. Coromel, « A generalized model for concurrent and distributed object-oriented programming », *Proceedings of the ACM SIGPLAN Workshop on Object-Based Concurrent Programming*, 1988.

7. J. Desbiens, M. Janta-Polczynski et G. Nault,« Projet Aurora: Systèmes Experts parallèles appliqués à la simulation de l'équipage d'un avion militaire », Actes du colloque EXPERSYS'92, Paris, France, 1992, pages 97-101.

8. D. Gagné, G. Nault, A. Garant et J. Desbiens « Aurora: A Multi-Agents Prototype Modelling Crew Interpersonal Communications Network », à paraître dans les actes de l'*Atelier du MDN sur les systèmes à base de connaissances et la robotique*, novembre 1993.

9. P.B. Hansen, « A Keynote Address on Concurrent Programming », *Computer*, vol. 12, 1979, pages 50-56.

10. P. Henderson, « Functionnal Programming: Applications and Implications », Prentice-Hall, Englewoods Cliffs, New-Jersey, 1979.

11. C.E. Hewit, « Planner: A language for Proving Theorems in Robots », *Proceedings of the 1st IJCAI*, Washington, D.C., 1969, pages 295-301.

12. C.E. Hewit, P. Bishop et R. Steiger, « A Universal Modular ACTOR Formalism for Artificial Intelligence », *Proceedings of the 3rd IJCAI*, Stanford, California, 1973, pages 235-245.

13. C.E. Hewit, « Control Structure as patterns of passing Messages », in "Artificial Intelligence: an MIT Perspective", vol. 2, The MIT Press, Cambridge, Massachusetts, 1979.

14. C.E. Hewit et H. Lieberman, « Design Issues in Parallel Architectures for Artificial Intelligence », AI Memo 750, AI Lab, MIT, Cambridge, Massachusetts, 1979.

15. C.A.R. Hoare, « Communicating Sequential Processes », *C.A.C.M.*, vol. 21, n° 8, 1978, pages 666-677.

16. Y. Ishiwkawa et M. Tokoro, « A Concurrent Object Knowledge Representation Language Orient84/K: Its Features and Implementation », *OOPSLA'86 Proceedings*, ACM, New York, 1986, pages 232-241.

17. D. Kafura et K. Lee, « ACT++: Building a Concurrent C++ with Actors », *Journal of Oriented Object Programming*, May/June, 1990, pages 25-37.

19. H. Lieberman, « Using prototypical objects to implement shared behavior in object oriented systems », *Proceedings of the First ACM Conference on Object-Oriented Programming Systems, language, and Applications*, Portland, Oregon, 1986, pages 214-223.

20. G. Massini, A. Napoli, D. Colnet, D. Léonard et K. Tombre, « Les langages à objets: Langages de classes, langages de frame, langages d'acteurs », InterEditions, Paris, 1989.

20. P. Naur, « Revised Report on the Algorithmic Language ALGOL 60 », *C.A.C.M.*, vol. 6, n° 1, 1963, pages 1-17.

21. L.A. Stein, « Delegation is inheritance », *OOPSLA'87 Conference Proceedings*, publié dans *SIGPLAN Notices*, vol. 22, 1987, pages 138-146.

22. L.A. Stein, H. Lieberman et D. Hungar, « A shared view of sharing: The Treaty of Orlando », in "Object-Oriented Concepts, Applications, and Databases", Addison-Wesley, Reading, Massachusetts, 1988.

23. B. Stroustrup, « The C++ Programming language», Addison-Wesley Company, Reading, Massachusets, 1986.

24. D. Ungar et R.B. Smith, « SELF: The power of simplicity », *OOPSLA'87 Conference Proceedings*. publié dans *SIGPLAN Notices*, vol. 22, 1987, pages 227-241.

25. A. Yonezawa et T. Watanabe, « Object-Oriented Concurrent Programming », The MIT Press, Cambridge, Massachusetts, 1987

26. « CLIPS Reference Manual, volume 1, Basic Programming Guide, CLIPS version 5.0 », January 11th 1991, Software Technology Branch, Lyndon B. Johnson Space Center.

27. « CLIPS Reference Manual, volume 2, Advanced Programming Guide, CLIPS version 5.0 », January 11th 1991, Software Technology Branch, Lyndon B. Johnson Space Center.

905450

Knowledge Representation Issues in Blackboards for ESM

Gregory Bourassa[1]
Lockheed Canada Inc.
1 Iber Road
Stittsville, Ontario
K2S 1E6

#149731

Abstract

A prototype blackboard system for Electronic Warfare Support Measures (ESM) systems is now in use to permit problem solving by multiple interacting "experts" (Expert Systems). These experts take the form of concurrent tasks in the Actor model. The system supports the representation of measured data, events, and event sequencing. It also supports uncertainty representation through the Dempster-Shafer model of evidence. The expert knowledge is currently represented procedurally in the algorithms executed by the actors in response to events.

The generic blackboard system is customized to the problem domain through the use of multiple named levels, dedicated to particular events and hypotheses. The entities residing on the various levels must nevertheless be interconnected to support the propagation of evidence and adjustment of beliefs.

The design of the blackboard system, belief system and experts is outlined, along with the customizations to the problem domain. The issues surrounding the uniform handling of facts, hypotheses and events are examined, as are those surrounding the propagation of evidence. Problems encountered with search algorithm complexity and with the representation of expert knowledge are presented, and a summary is made of lessons learned.

Introduction

Prototypical blackboard systems for Electronic Warfare Support Measures (ESM) systems have been developed by both Defence Research Establishment Ottawa (DREO) and Lockheed Canada. Through these developments and the integration of the prototypes with larger ESM software systems, experience has been gained in the utility of blackboard systems applied to various areas of the ESM problem. This experience has also proved useful in confronting the design and implementation problems connected with blackboard systems in general.

This paper will focus on issues of knowledge representation. It is not intended as a tutorial on ESM systems nor on blackboard system architectures, so we will provide the relevant facets of these in cursory overview. This will be sufficient, it is hoped, to provide the reader with the necessary context for understanding the knowledge representation issues.

Electronic Warfare Support Measures (ESM) Systems

ESM systems consist of radar antennae and receivers (but not emitters) and associated processors and operator consoles. Their mission is to "listen" to the radar environment in its full frequency bandwidth and in an omnidirectional fashion, to the extent that technology will permit. The operational value of this is to provide a full view of the Electronic Warfare (EW) tactical environment without radiating any signal. ESM systems enjoy, in principle, a range advantage over radar seekers, given equal sensitivity, because they receive the signal after it traverses the path between the seeker and the ESM's platform only once. The seeker must receive the return signal reflected by the ESM's platform (i.e. two path traversals). This signal is thus substantially attenuated, as compared to that received by the ESM system. This translates into an ability of the ESM system to "see" an emitter before the emitter can detect the ESM system's platform. See Figure 1.

---

[1] Phone: (613) 836-6860; Fax: (613) 836-6566

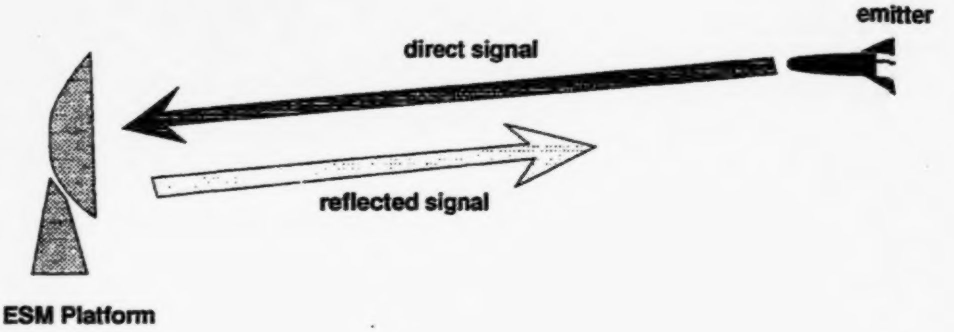


Figure 1.
ESM range advantage

On the other hand, ESM systems can be at a disadvantage in that they must, in real time, interpret arbitrarily complex environments, with interleaved pulse trains from various emitters. They must accomplish this without the prior knowledge (which active have) of the frequency, pulse width, interpulse time delays, etc., of the signals. They face, in effect, the ultimate open-ended feature extraction and pattern recognition problem. Modern ESM systems attempt to provide the operator at the ESM console with the identities of all the signals in the environment (emitter types), some information about their possible platforms, and access to much of the raw data and intermediate information developed in arriving at these identifications. New systems under development also attempt to provide higher level inferences about the tactical significance of the current environment.

The processing necessary to deinterleave and interpret the signals in the environment is done by a mixture of specialized filtering hardware, and software running on general purpose computers embedded in the ESM system.

Blackboard Systems

It is in this latter area (embedded system software) that blackboards have been applied. In particular, they form the basis of experimental Tactical Situation Analysis (TSA) subsystems. Blackboard systems are a form of software architecture. They provide a relatively new alternative within the classical "divide and conquer" approach to handling complex problems.

The blackboard analogy is as follows: A common blackboard is shared by a group of experts and workers. They may not always work at the same time, and in any case they are not permitted to communicate with each other except by reading from, writing to, or erasing (in whole or in part) the blackboard. The workers may place new facts on the blackboard and otherwise maintain it, but may not comment nor solve problems. The experts each have areas of specialty, and are attempting to contribute this to the team solution of a problem. One person, a controller, provides blackboard access to each reader and writer, possibly using a priority scheme. Each person having access is governed by rules of conduct, which differ in various implementations of the blackboard paradigm. Various persons may be allowed to read but not write, to write anywhere but not erase, to modify at will, to write only in certain areas or about certain topics, etc., etc. The controller may also notify experts that something in which they are interested has been posted on the board (providing they told the controller of their interest in these particular events).

The software implementation of this scheme (see Figure 2) permits the problem area to be subdivided into internally cohesive modules representing the various experts and workers. It also simplifies application architectures by reducing the complexity of the control flows that appear in the algorithms. The complexity of control flow exists but migrates to the generic blackboard controller object, and away from the application algorithms. There is a single, structured data area and the various modules are invoked by the controller based on the arrival of data ("blackboard events") relevant to those modules. Thus, the designer need not foresee every control flow ordering (provided the appropriate states are handled graceful), and the system can respond appropriately to incoming data. This represents a potential improvement over simple polling schemes, wherein the modules are invoked in a predetermined order, whether they are relevant to the current situation or not. Blackboards are particularly valuable where no definitive "solution" or overall processing "goal" is easily definable, but where an evolving state space can adequately represent the evolving view of the situation. This property of blackboards makes them a very natural choice for TSA subsystems, amongst others in ESM.

Types of Knowledge to be Represented

We can now proceed to set the stage for the remainder of the discussion by defining the kinds of knowledge we need to represent on (or associated with) ESM blackboards. We assume the TSA subsystem case, and base our discussion on Lockheed Canada's version of this subsystem except where the DREO model is specifically mentioned.

First, we need to represent facts, or our nearest approximation to them, in the form of measurements made by the ESM receiver and its associated filtering and preprocessing systems. These measurements we can call tracks, where a track is taken to represent one operational mode (a signal format over time) of one emitter. These tracks carry a variety of parameter values, and a prioritized list of possible identifications (IDs) along with the

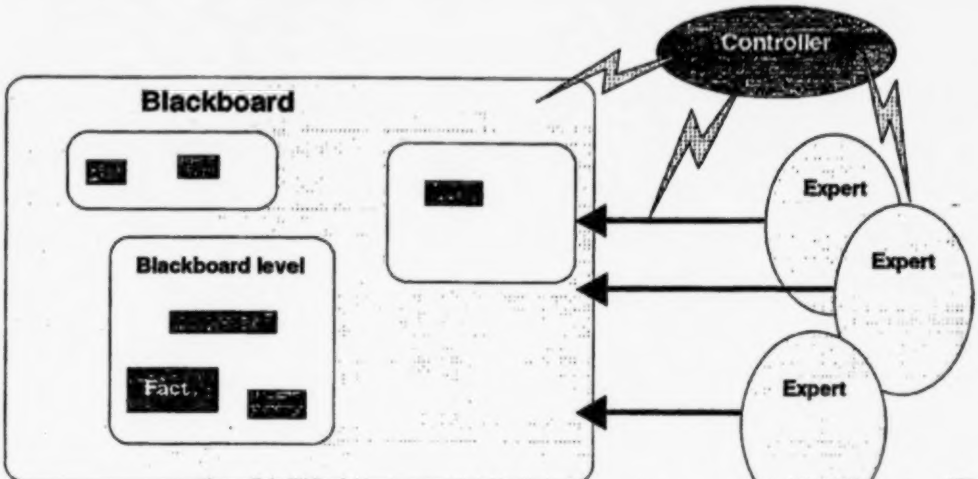


Figure 2.
Blackboard Architecture

belief associated with each candidate identification. We immediately note the fuzzy nature of our data, and the potentially shifting sand on which the rest of our structures and inferences must be built. The ordering of the system's list of IDs for a track could possibly be changed by the ESM front-end based on new data arrivals, at any time.

The second kind of knowledge which must be represented consists of the results of making inferences about tracks. We will refer to these entities as hypotheses. They can include:

a) hypotheses which correlate multiple tracks as modes of the same emitter;
b) hypotheses which correlate multiple tracks and construe them to represent the same emitter at different times and /or places;
c) hypotheses which group multiple emitters on an inferred platform type, based on archival knowledge about typical emitter fitments of various platform types; and
d) hypotheses which are developed by using very precise measurements and intelligence data to construe a given emitter or platform to be a specific known one (e.g. not just a Patrol Frigate, but the actual name of the individual ship).

A third type of knowledge which must be represented is scenario or script knowledge about the typical tactical behaviour of emitters and platforms. This kind of knowledge includes representations of *expected* events, based on the as yet unfulfilled (future) portions of the script. This, as we will see, raises the whole issue of retracting failed expectations and their hypotheses, and attempting to restore a consistent previous state (or establish a consistent new one).

A fourth kind of knowledge is the "expertise" required to make inferences about the facts and hypotheses residing in the blackboard data store. This knowledge can be represented as rules or procedures, and is stored separately from the main data stores of the blackboard, within the scope of the individual blackboard "experts". The side-effects of making inferences with this knowledge are the creation, modification and destruction of the hypotheses on the blackboard, and the modification of evidence values.

These four constitute the main types of knowledge which we will discuss in this paper. In the following sections we will examine the design choices made for the blackboards and the experts, and highlight their strengths, weaknesses and the lessons learned with particular emphasis on the role of the knowledge structures.

The Design of the Prototype

Generic Blackboards

The prototypes were implemented in Smalltalk/V. In the DREO case the prototype was built on a framework of custom-built, generic blackboard classes. This part of the work proved to be very valuable, and allowed the ESM application-specific portions to be implemented separately from those parts which represented archetypical blackboard functionality. Included in the generic portions are a blackboard class which facilitates the definition of multiple named blackboard "levels". These levels represent distinct data holding areas, and allow the material on the blackboard to be organized. Also included is a generic class for the blackboard controller. This controller class manages an event queue and allows external experts to register for named events (the event names and priorities also can be loaded when the blackboard is initialized). The controller then invokes these experts only when events of interest to them arrive in the blackboard event queue. These events, which may contain relevant input data, can be posted to the blackboard by any expert or worker process.

The generic classes do not preclude the implementer from creating specialized blackboard-object classes which exhibit application-specific behaviour. The data areas can hold any kind of objects, and even mixtures of disparate classes of objects. In fact, specialized TSA blackboard object classes were created for both of the prototypes. These blackboard objects encapsulate knowledge and behaviour related to the ESM problem area, and the TSA in particular.

Dempster-Shafer Belief System

Uncertainty is managed through the Dempster-Shafer system of evidence [1]. The scheme is tailored to the case of competing alternatives that can be represented in a tree topology, using optimizations derived from work by Shortliffe and by Logan [2]. This is meant to allow relatively good performance when beliefs must be recalculated.

The performance is known to be good relative to that for the full implementation of the Dempster-Shafer system. Its real-time adequacy is not known but will be tested as the DREO prototype system approaches a more finished and deployable state.

The system of evidence requires that the weight of evidence associated with facts (track information, in this case) be distributed amongst, or attributed to, the various hypotheses that are built over the track. For example, if there are several candidate emitter IDs, there will be an emitter hypothesis for each of the candidate IDs [3]. As we will see below, this has an impact on the storage representation and organization of the knowledge on the blackboard. Most of the objects on the blackboard must be interconnected with each other to support the evidence system.

Each track fact is attached to two belief hierarchies, one for emitter hypotheses and one for platform hypotheses. Mechanisms exist to amalgamate the beliefs from the two hierarchies within one track, and to deal with beliefs across several tracks. Interdependencies are also required to handle the adjustment of beliefs based on the hypothetical grouping of one or more emitters as fitments of a single platform. This requires novel features for the flow of the weight of evidence both up and down the hierarchy of hypotheses. Some of the features are heuristic, but it is believed that the performance so far adheres quite closely with that of the more formal mathematical models.

Instantiation of Blackboards for the Prototype

The blackboard system for the prototype is instantiated with three levels and a list of events with priorities. The levels are: track, emitter, and platform. The experts can register to be informed of events, and to be given control of the blackboards as a consequence. Areas of expertise include, but are not limited to, the emitter modes (emitter to emitter correlation), emitter fitments (emitter to platform correlation), spatio-temporal reasoning (platform to platform correlation), and a hull identification module (applied currently only in platform to platform correlation, due to the absence of a hull identification database). Most of these experts would, for example, register to be informed of a NewTrack event.

When a new track is placed in the track level of the blackboard several operations ensue. Emitter hypotheses are created for each of its candidate emitter IDs, placed on the emitter level of the blackboard and linked to the track in a belief hierarchy. Platform hypotheses are created from platform emitter fitment lists (from library information) for each of the platforms to which one of the candidate emitters can be fitted. These are placed on the platform level of the blackboard and another belief hierarchy is attached to the track to connect these platform. Later, emitter hypotheses themselves can be arranged hierarchically, as can platform hypotheses (see Figure 3).

We now have at least one track on the track level, and possibly several emitter hypotheses and several platform hypotheses on their respective levels. These items are all connected to the track, but not to each other. They can become linked to each other when expertise is applied to correlated two tracks as one emitter, or many emitters as members of a fitment on one platform, etc. In such cases a hierarchy of hypotheses is created, and beliefs redistributed using the linkages.

Each of the assertions of new blackboard entities gives rise to a blackboard event notification which can invoke experts to do additional work. In the above discussion we would expect at least one NewTrack event, and possibly several each of NewEmitter and NewPlatform events.

Each track fact and hypothesis is maintained as a specialized blackboard object, held in a blackboard level and capable of holding or pointing to evidence objects and the other blackboard objects to which it is related. The blackboard objects belong to a common hierarchy, and are specialized through subclassing.

Experts (Representation and Capability)

The experts encapsulate knowledge about how tracks can be correlated to each other (as, for example, modes of one emitter), how emitters can be correlated to platforms, how platforms can be correlated to each other over time, and so on. They contain the majority of the ESM expertise, while the blackboard objects contain measurements and place-holders for inferences (hypotheses).

The experts are implemented as Actors, the equivalent of concurrent tasks. This is to permit future porting of the software to parallel hardware where the Actors can exhibit true concurrent

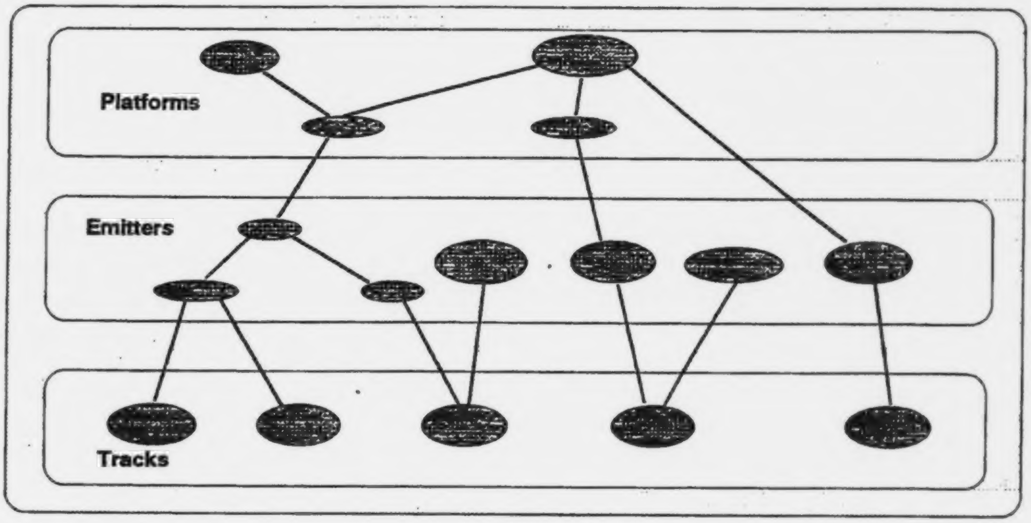


Figure 3.
A hierarchy of evidence on the Blackboard
(Track-Platform connections not shown)

behaviour, rather than time-slicing or other scheduling. At the moment, the software is run on a single workstation and the concurrent behaviour simulated by the Smalltalk/V system. Once an Actor obtains control of the blackboard, it runs to completion of its task unless it specifically invokes another Actor.

Whether implemented as Actors or merely objects, the scheduling scheme is the same, and adheres to the blackboard model. The controller takes a new event from the blackboard event queue and notifies all of the experts, in turn, which have registered as having an interest in this kind of event. As each expert is notified, it has the opportunity to take control of the blackboard and perform processing in response to the event. Typical processing includes reading relevant information (cf: knowledge) from the blackboard, asserting new hypotheses onto the blackboard, performing or invoking the belief system adjustments, and posting new blackboard events to the queue.

The knowledge in these experts is not represented as objects. It is embedded in the procedural code which implements the methods. These methods are called to handle an event, and the code is executed. These methods typically call other methods in the same and other classes. Thus the code embodying a particular piece of expertise can be distributed over several instance methods, potentially in several different classes.

The sophistication of the expertise so far captured in this way is relatively modest, partly because of the very limited number of inferences that can safely be made based on candidate emitter IDs and fitment lists. Perhaps the most interesting areas of expertise involve the hull identification module and the still rudimentary spatio-temporal reasoning. Another very interesting area is that of the scripted knowledge, but this is the least developed area of the system and has only been tried off-line from the main prototype.

The hull identification module uses precise measurements of Pulse Repetition Interval (PRI). The current system uses this information to correlate platform hypotheses to other (earlier) platform hypotheses.

The spatio-temporal reasoning involves the expression of some rough constraints on motion over time, based on the inferred (hypothesized) platform type's maximum velocity and the elapsed time. It is used to prevent falsely correlating similar platforms as being the same platform at different times. It can prevent such false correlations when the spatial constraints are exceeded for the elapsed time between the observations. A very rough approximation of the range of the emitter is also developed, based on archival knowledge about emitted power for this emitter type, along with some very conservative defaults.

The script knowledge is captured in the form of script objects. It can be used to attempt to set up hypotheses representing expected events, based upon fulfilment of the early portions of the script. It may become useful in anticipating the various parts of missile launch and homing scenarios. It has not as yet been extensively tested, and is not integrated with the full ESM system. It will rely heavily on pattern matching the unfolding state of the blackboard against the expected state from the script. This level of pattern matching may be difficult to express in Smalltalk methods.

Experiences

So far, we have described the outlines of the blackboard system for ESM, the kinds of knowledge stored and the method of its storage. We now proceed to make observations on the experiences arising from the implementation and exercise of these structures.

Problem Solutions and Partial Solution States

The blackboard system has proved a very flexible and powerful tool for the expression of evolving or partial solutions. In this context, we refer to an interpretation of the radar environment and its implied tactical situation as the solution. Since the degree of uncertainty about the meaning of each and every measurement is rather high, we can not easily define a threshold at which a single acceptable solution is achieved. Hence the blackboard contents, taken as a whole, are used to represent the current solution. The contained information can be displayed to the operator as a suggested interpretation only.

Retention of Multiple Candidate Solutions

An interesting implication of viewing the blackboard contents as a proposed solution is that we maintain many, possibly contradictory and competing, candidate solutions. If we have several emitters for a track, and several platforms which might carry one or more of these emitters on each, then we have a multiplicity of possible views of the environment already. Compounding this situation by many tracks makes the level of complexity very great indeed. Some method is required of organizing all of this information for two reasons. First, it must be presented to the user in some reduced and relatively simple form. Second, it must be capable of being adjusted consistently as new information arrives or old information is controverted and discarded.

The Handling of Uncertainty

The Dempster-Shafer system of evidence helps to organize and prioritize all of this information, at the expense of some processing time and some complex cross-linkages amongst the information. It also provides a consistent method of reorganizing and reprioritizing the information based on the arrival of new information, and of handling the fact that some uncertainty is associated with all facts and hypotheses, new and old. This has proved to be a strong point in the overall representation of the knowledge, and it is hoped that it will be amenable to tuning for real-time performance.

Outstanding Issues

Here we advance some observations on areas that still require significant work, either in the form of additional research and development, or of redesign. Most of the observations arise from work done with the Lockheed Canada prototype, but are general enough to be applied to any similar blackboard implementation for ESM.

Presentation of Multiple World Views

While we now have a very powerful overall capability to retain and organize multiple world views on a single blackboard (based on multiple hypotheses for each track, and on hierarchies of hypotheses), the world views themselves are implicit in the belief trees and difficult to present. It would be nearly unintelligible to the operator to present all of the hypotheses and facts at once in some tabular or graphical display. Alternatively, it is probably misleading to present only the most believed hypotheses about the emitter and platform for each track.

We have discussed various graphical representations of entities that are uniquely believed versus those that have alternatives. This might allow the user to examine the alternatives at will, or to accept the system's best interpretation.

Another approach might be to allow the user to constrain the presentation by choosing their preferred hypotheses one at a time. As one hypothesis is chosen, the system could suppress the display of the competing hypotheses in the belief tree, and prune any other dependent parts of the world view. The user could then perform the same operation on a remaining hypothesis for which there are alternative; the system would prune the world view accordingly, and so on.

Here the knowledge representation problem reduces to a knowledge display problem. If we are content with our internal representation for competing hypotheses, the constraints set by the user really do not change it, but rather change the way we traverse it to display remaining alternatives. Because the highest-rated hypotheses may be suppressed by the user, however, the traversal may become very complex, and new knowledge representations will probably have to be developed to guide the search.

Cross Level Linkages in the Blackboard

While the Dempster-Shafer system is acknowledged to be very consistent and well supported by theory (notwithstanding the approximations and heuristics we have introduced), it does introduce a somewhat undesirable artifact in the form of the linkages amongst objects on the various blackboard levels. These linkages are used for belief tree traversals, and are fortunately very easy to express in Smalltalk (wherein adding an object to some other object's list of things is really translated into the storage of a pointer, and the pointer complexity is handled by the language). Where they really do complicate matters is during the assertion of new hypotheses, and even more so during the retraction of obsolete hypotheses. The management of this adds considerably to the code and makes the blackboard organization less simple and robust.

Complexity and Uniformity Issues for Knowledge structures

Smalltalk is an excellent vehicle for the implementation of objects, encapsulating their own data and behaviour. It is thus very useful for representing facts, along with all the functions necessary to manage the housekeeping for the facts. This has made the complexity of the belief structures feasible. In many other languages, the TSA module as it stands would have been prohibitively expensive to implement based on the belief structures alone.

When one extends the notion of facts (relations amongst ground data entities or groups of them) to that of rules (relations which are defined in terms of other relations), in order to capture knowledge, the picture becomes complicated again. It is desirable to express rules and facts in a basically uniform manner as is done in various expert system shells, and more powerfully and generally in declarative programming languages such as Prolog [4]. This permits them to be both expressed and evaluated uniformly.

Unfortunately, Smalltalk it is not a declarative language, nor does it provide any native features that facilitate the declarative expression of knowledge. For this reason, the facts cannot be construed as degenerate forms of rules, as is the case in logic programming environments and modern knowledge engineering tools. The rules must be implemented as methods (e.g., procedures). The unfortunate result of this is that the expert knowledge is dispersed among many methods in several classes, and is difficult to examine, debug, enhance and maintain. The Prolog/V classes in Smalltalk were tried as a means of improving this situation, but proved to be of only tutorial interest. This little emulation of Prolog which is delivered with Smalltalk/V has no tail recursion elimination (last call optimization) to protect the Smalltalk stack, and only a very restricted capability to pattern match on native Smalltalk objects [5].

The Visibility and Impact of Solution-Space Search Algorithms

Because neither declarative versions of expert rules nor built-in search nor pattern matching are provided in the Smalltalk environment, the blackboard experts also suffer from a heavy responsibility for managing their own search strategies on the blackboard. More lines of code are dedicated to this pursuit within the expert methods than are dedicated to accomplishing the inferences themselves (that is, to representing the expertise). In a typical example drawn from the DREO prototype, most of the code in a method for trying to make a platform-to-platform inference is given over to data search control, featuring four levels of nested code blocks, three of them to scan through various collection-oriented parts of the blackboard.

This is not meant to imply that such code is wrong. It is just expensive to write and maintain, and vulnerable to human error. A simplified and more declarative facility for capturing expert knowledge is needed.

Summary

Lessons Learned and Future Directions

The development of a TSA knowledge-based module for ESM has been very successful. The blackboard implementation has proved extremely practical and powerful. The representation of knowledge in the form of facts is well supported by Smalltalk, as is the interconnection of facts as necessary to support belief trees. The representation of alternative world-views to the user needs more study, and may need new knowledge structures to manage it successfully. The representation of expert knowledge within the blackboard experts is a weak point, being arduous and more heavily loaded with control strategy code than knowledge structures.

An issue which deserves further consideration is the integration of modules in disparate languages. With the advent of shared libraries in the Unix world and Dynamic Link Libraries (DLLs) in the OS/2 and Windows world [6], the bias towards unilingual programming has been reversed and suiting the tool to the task (or sub-task) is becoming the norm. Libraries or processes created in any language should be able to communicate with each other. Providing a similar compatibility (even at the level of object modules which can be statically linked to each other) for the embedded processors for ESM should be a priority. With it, addition of expertise to the blackboard experts could be accomplished with clarity, precision and speed.

Acknowledgements

We gratefully acknowledge our debt to the following organizations and individuals: DND and DREO for their continued support of research into ESM algorithms; DREO for the permission to read the source code of their prototype TSA subsystem to verify the accuracy and coverage of this paper; John Altoft for his design and implementation work on blackboards and TSA subsystems both at Lockheed Canada and DREO; Selim Ulug and Software Kinetics Limited for work on generic blackboard classes; Dr. Arthur G. Self and Lockheed Canada Inc. for encouraging the writing of this paper.

References

1. Shafer, G., *A Mathematical Theory of Evidence* (Princeton University Press, Princeton, NJ, 1976).

2. Shafer, G., and Logan, R., Implementing Dempster's Rule for Hierarchical Evidence, in: *Artificial Intelligence*, 33, (1987) 271-298.

3. *Blackboard Systems for ESM*, (Lockheed Canada document).

4. Clocksin, W.F., and Mellish, C.S., *Programming in Prolog*, (Springer-Verlag, ISBN 0-387-17539-3).

5. *Smalltalk/V Tutorial and Programming Handbook*, (Digitalk Inc., Los Angeles CA).

6. H.M. Deitel an M.S. Kogan, *The Design of OS/2*, (Addison-Wesley, ISBN 0-201-54889-5).

905C20

Knowledge Acquisition and Conceptual Design for a TACAN Siting Advisor System

William H. Henneker and C. S. Phan
Knowledge Systems Laboratory, Institute for Information Technology
National Research Council of Canada (NRC), Ottawa, Ontario, K1A 0R8
Phone: 613-993-8556, Fax: 613-952-7151, E-mail: bill@ai.iit.nrc.ca

#149733

Abstract

The siting of ground-based Tactical Air Navigation (TACAN) system antennas is a complex planning task. It involves compromises among many competing factors such as navigational aid requirements, aerodrome safety zone criteria, interference due to multipath propagation effects of radio waves, radio frequency coverage of the TACAN's signal and costs associated with preparing a site and installing the antenna.

The Knowledge Systems Laboratory of NRC and the Directorate of Communications, Electronics Engineering and Maintenance (DCEEM) of Department of National Defence (DND) are collaborating to develop a computer-based advisor system to assist in the siting of TACAN antennas. The project is being funded by Chief of Research and Development (CRAD) and DCEEM is providing the services of a TACAN siting expert from whom knowledge is being acquired. Personnel from DCEEM will be the users of the prototype system and will validate it in a future siting task.

The proposed computer-based advisor includes a TACAN siting tool, a knowledge base, a knowledge browser, a siting assistant and a tutor. Each of these modules will be discussed in the paper. The paper also discusses the strategy for acquiring knowledge from the TACAN expert.

Introduction

An expert system is a software module that embodies the knowledge of an expert. Most expert systems attempt to solve a specific problem or perform a specific task on behalf of a user. An advisor system also embodies the knowledge of an expert. On the other hand, it is a software module which, when coupled to an interactive computer application, can assist the user in solving a problem or performing a task. It can act as a tutor introducing the beginner to the application. It can act as a consultant providing help by responding to questions asked by the user, supplying advice and support. It can also monitor the user's interaction with the application and intervene with advice aimed at allowing the user to achieve the desired goal more effectively.

The development reported in this paper is based upon an architecture for advisor systems comprising three advisor functions: Proactive Help, Reactive Help, and Tutorial[1,2]. The prototype implementation of this system architecture has been reported for the application domain of Entity-Relationship Modelling and a system called ERMA (Entity Relationship Modelling Advisor) has been developed[2].

The current application domain is the siting of TACAN antennas for DND. The siting of such antennas is a complex planning task. For the past 18 years, DND has relied upon the services of an expert who has spent a lifetime of involvement with radio frequency navigational aids. DND would like to capture the expertise of this individual in an advisor system that would assist new personnel to learn and then perform the siting task.

The current application domain provides an additional, significant challenge to the advisor system builder because of the wide variety of knowledge that must be included in the system and the quantitative methods used to evaluate the suitability of a particular site.

Nature of the Siting Task

TACAN is a military air navigation system. It provides bearing and distance information from a reference point — the TACAN transmitting antenna — by which a pilot with a suitably equipped aircraft can fix his geographical location[3]. DND is currently responsible for 58 TACAN or combined Very High Frequency Omnidirectional Radio Range (VOR)-TACAN facilities. The implications of this responsibility are: resiting of existing installations for improved performance, diagnosis of problems at existing facilities and the siting of new facilities. The DND expert has provided his services for the siting of at least 16 of the current 58 installations. Each siting task involves a visit (maximum of ten days) to the specific region in which the antenna will be installed and the preparation of a report giving the geographic coordinates of the antenna's proposed location, its height above immediate terrain and the type of antenna to be installed.

A human performing the siting task must attempt to minimize the convergence angles of the approach radials to the runways being serviced by the TACAN. Constraints on this heuristic minimization process include certain airport safety zone criteria, interference due to multipath propagation effects of radio waves, distance limits (or radio frequency coverage) of the TACAN's transmitted signal, availability of real estate and costs associated with preparing a site and installing the antenna.

Knowledge Domains

The antenna siting task requires specific knowledge that must be obtained from the following generic domains.

TACAN System Principles

TACAN is a radio frequency navigational aid with operational frequencies in the range of 960-1215 MHz. Superimposed upon this carrier frequency are 15 and 135 Hz modulations[3]. These modulations are created by an antenna design that includes rotating inner and outer cylinders with embedded parasitic elements. The rate of rotation of the cylinders is 900 rpm. These modulations carry the bearing information in the form of a north reference pulse that is emitted once every rotation of the cylinders. Of particular importance to the siting expert is the vertical directivity curve of a specific antenna. This curve provides a measure of normalized signal level as a function of elevation angle and is important in assessing multipath propagation characteristics of a TACAN antenna.

Radio Frequency Wave Propagation

Of particular interest to the siting expert are variations in the composite radio frequency signal received by a suitably equipped aircraft due to multiple paths by which the signal reaches the aircraft. Two of the most important factors in creating multiple paths are specular or ground reflections and vertical or side reflections. The former case is shown in Fig. 1 where the resultant signal received by the aircraft is composed of a direct signal and one reflected from flat terrain. Idealized mathematical models exist for evaluating the effects of such reflections. In this case both the direct and reflected rays carry the same azimuth information. Figure 2 shows a plan view in which the vertical surface of a nearby building, composed of a highly reflective metallic substance, produces a side reflection which combines at point B with the direct signal emitted by a TACAN antenna. In this case, both the direct and reflected rays carry different azimuth information. If, for some reason, the signal level of the direct ray is attenuated, either by an obstruction placed between the receiving and the emitting antenna or by interference due to specular reflections, the ray created by the building's vertical surface may take over control of the aircraft's receiver

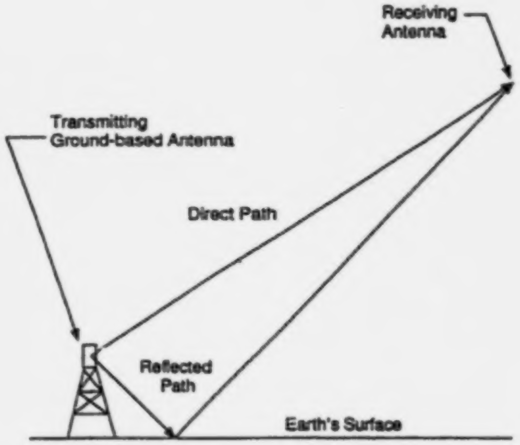


FIGURE 1. Elevation View Showing Two Path Interference Phenomena.

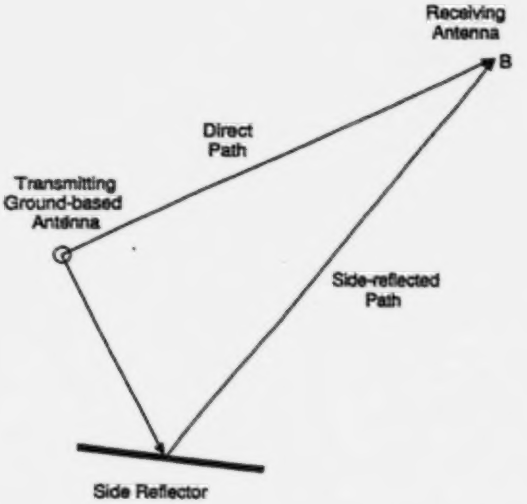


FIGURE 2. Plan View Showing Geometry of a Side-reflected Signal.

causing the aircraft's azimuth indicator to indicate an incorrect bearing or to oscillate.

Airport Zoning Criteria

This knowledge is classified under the generic title of design rules. It includes information such as the minimum displacement of obstructions from active runways or taxiways and the prescribed heights of obstructions that are placed at locations greater than the minimum displacement from active runways or taxiways. Also included in this general category of design rules are prescriptions governing the location of TACAN beacons relative to other navigational aids, fuel storage areas, ammunition storage areas and living quarter areas.

Map Reading and Manipulation

The siting expert is able to perform basic operations with topographic maps describing airports to scale 1:2,000. He is able to locate a point either in geographic coordinates (latitude and longitude), grid coordinates (Universal Transverse Mercator) or relative coordinates obtained from experimental observations on site. He is also able to draw lines of specific length and specific angle relative to other lines. The siting task also entails an ability to locate magnetic north on these base charts.

Surveying

The siting expert must collect a variety of data while he is on site. One of the most important pieces of experimental information is the data acquired by operating a theodolite which measures bearing and elevation data relative to the proposed TACAN site. The siting expert also has an appreciation for the procedure by which a theodolite is initially aligned so that its reference is magnetic north.

Air Traffic Navigation

Depending upon the operational requirements, the TACAN system can provide coverage for non precision approaches of aircraft to an aerodrome and enroute coverage for an aircraft flying from point A to point B. Basic to the siting task is a specification of approach radials along which an aircraft travels as it makes its approach to a runway. These approach radials are given in terms of their magnetic heading. The siting expert must also understand the concept of a let-down profile. This is a profile of altitude versus distance from threshold and is followed by an aircraft during its approach to a runway. The expert compares let-down profiles with other profiles called null-loci. These latter profiles depend upon the height of the TACAN antenna above immediate terrain. By varying the height of the antenna, the expert is able to optimize the location of the null-loci profile relative to the let-down profile. This ensures that the aircraft will always receive sufficient signal as it moves through space.

The siting expert routinely reads the appropriate enroute low altitude charts to ascertain if the coverage provided by the TACAN which he is siting is sufficient to fulfil the minimum enroute altitudes specified in these enroute charts.

Knowledge Acquisition

Domain expertise for this project is provided by a single expert, a very experienced individual who is an acknowledged expert on the subject of TACAN antennas. The expert uses various types of knowledge while assessing cases and preparing recommendations. These include:

- facts, e.g.,
  The control tower is 311.0 meters west of the runway centerline and 371.6 meters south of the runway's northern edge.
- classificatory knowledge, e.g.,
  RTA2 antennas are a type of TACAN antenna.
- mathematical and physical principles, e.g.,
  A circular beam will have an elliptical cross-section after reflection from a spherical surface.
- graphical analysis, e.g.,
  A beacon in any of the indicated areas would provide acceptable radio frequency coverage on all required approach radials.
- procedural knowledge, e.g.,
  The theodolite should be set to magnetic north.
- concepts and relations, e.g.,
  Faulty equipment and transient reflections can both produce signal degradation.
- heuristics, e.g.,
  Continuous loss of auxiliary pulses cannot be caused by the site environment.
- meta-knowledge, e.g.,
  The analysis requires more detail.

An important task of knowledge acquisition is to determine the expert's model of the solution. One mark of an expert is that he does not approach a problem with a 'blank-slate' as a novice would but has a private, internal representation of the domain[4,5]. This includes concepts or topics at various levels such as knowledge of key factors which have to be considered, a set of expectations about information that will be required, case histories stored in memory of previous problems and solution strategies, and requirements for report preparation. Another aspect of the expert's model is his working procedure which may vary from assignment to assignment and may be regarded as a library of outline plans to be filled out and implemented as appropriate.

How do we capture the expert's knowledge? The answer appears to be a combination of structured interviews led by knowledge engineers and case descriptions presented by the expert[6,7]. The case presentations permit the knowledge engineers to ascertain the expert's working procedures and the library of outline plans that he unconsciously accesses during the siting process. They also permit the knowledge engineers to assess any heuristics used by the expert while he is performing a particular task.

Conceptual Design

This is a complex situation and there is no single homogeneous problem-solving strategy involved but rather a series of strategies or tasks which, taken together, should result in a complete analysis of the signal requirements at a particular air base. A computerized application which is to provide useful support in this area will need to have a range of capacities with distinct modules performing specific tasks and with an overall structure to provide guidance and a productive approach to the situation.

Our approach to this problem is a TACAN advisor system based upon the architecture for advisors previously developed [1,2]. The advisor system will eventually cover all types of knowledge used by the expert and is composed of a siting component and an advisor component.

The siting component is designed to provide various analytical and visual tools to assist the user in performing a siting task. This component consists of two modules: the Basic Siting Toolkit and the Siting Assistant.

The Basic Siting Toolkit is a collection of analytical tools addressing four areas, namely, radio frequency wave propagation analysis, radio line-of-sight or coverage analysis, approach sector terrain profile analysis and let-down profile analysis.

The Siting Assistant is intended to be a topographic map processing tool. It will allow the user to view and edit the airport configuration, and to perform preliminary site selection based on operational requirements such as convergence angle and siting rules such as airport safety zoning criteria. The tool will also guide the user through or remind the user of the various procedures necessary to lead to the successful conclusion of a siting task. It is also intended that this tool will provide some support in the production of a final siting report.

The advisor component is designed to provide assistance to users in order to enhance their productivity during a site selection task. Three kinds of help are addressed in the system, namely, proactive help, reactive help and tutoring help.

Proactive help is assistance that is initiated by the system. The system monitors user interactions to detect any constraint violations, mal-plans or misconceptions and intervenes to provide comments and suggestions if deemed appropriate. Proactive help is not a separate module but is integrated into the siting component.

Reactive help is assistance that is explicitly initiated by the user. This is accomplished by some explicit questioning mechanisms. We differentiate two kinds of questions, namely, static and context-sensitive questions. Context-sensitive questions are questions related to the current problem solving state. An example of such a question is: "what if I raise the antenna from its current height to 32 feet". Context sensitive questions are also integrated into the siting component. Static questions are general questions. Two typical types are: "Tell me about...?" and "How do I do ...?". Static questions are handled by a hypertext-style Knowledge Browser.

Tutoring help is intended to provide a coaching function. The system will guide the user to solve prestored siting cases. In general, it invokes the Siting Assistant to perform site selection simulations, calls the Knowledge Browser to explain concepts and procedures and the Basic Siting Toolkit to perform site analysis.

Figure 3 shows a top-level conceptual decomposition of the TACAN Advisor. The system is composed of 5 main modules: the Basic Siting Toolkit, the Siting Assistant, the Knowledge Browser, the Tutor and the Knowledge Base. A user may invoke any of the modules except the Knowledge Base. The four callable modules also provide mutual internal support to each other. All domain knowledge including the library of siting cases for the tutor will be stored in the Knowledge Base. It is the intent of this design that the system is to be implemented in such a way that future modifications to the Knowledge Base will not necessitate any changes to the other four modules.

Summary and Plans for the Future

The TACAN Advisor project was initiated in April 1992 and its first year involved five intensive

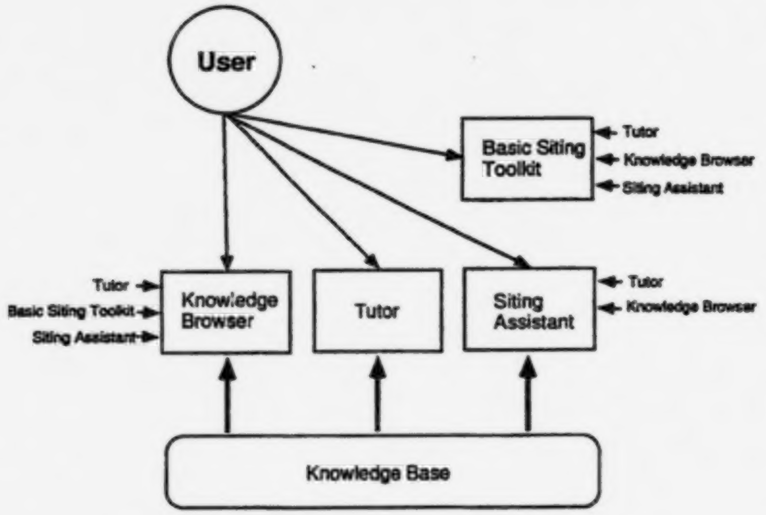


FIGURE 3. TACAN Advisor Conceptual Model.

knowledge acquisition sessions, each of approximately one week's duration. The developments discussed in this paper represent a first report concerning the project.

During the current fiscal year, NRC personnel will be implementing the Basic Siting Tool, developing content for the tutor module and investigating various topographic map processing packages with a view to purchasing software that will act as a platform for the implementation of the Siting Assistant. Of course, the software, itself, is not sufficient. Topographic maps to a scale of 1:2,000 in electronic format for the aerodrome environments discussed during the case studies presented by the domain expert will also be obtained. These topographic maps will be an integral part of the system's knowledge base and will permit the system's potential users to make graphical inferences similar to those made by the expert. They include Winnipeg, Ottawa, Iqaluit, Val d'Or and Greenwood in Nova Scotia. The conceptual design of the knowledge browser will also be considered during this fiscal year.

Tasks to be accomplished during the third and final year of the project include implementation of the Tutor and the Knowledge Browser.

Acknowledgments

The authors acknowledge the assistance of Mr. S.V. Greco, who is the domain expert from OCERG Navtronics, and Captain D.L. Stimson and Mr. P. Johnson of DCEEM during the first year of this project.

References

1. M.-M. Boulet, J. W. Brahan, A. J. Cole, P. Duchastel, J. R. Hartley, R. A. Orchard, A. T. Parent, R. Pilkington and C. S. Phan, "A Design Task Advisor", Artificial Intelligence and Education, *Proceedings of the 4th International Conference on AI and Education*, Amsterdam, The Netherlands. May 24–26, 1989. Edited by D. Bierman, J. Breuker and J. Sandberg, pp. 25–31.

2. J. W. Brahan, B. Farley, R. A. Orchard, A. Parent and C. S. Phan, "A Designer's Consultant", "Research and Development in Expert Systems IX", *Proceedings of Expert Systems '92*, The Twelfth Annual Technical Conference of the British Computer Society Specialist Group on Expert Systems, Cambridge, U.K. December 15–17, 1992, pp. 197–207.

3. S. V. Greco and H. R. Reed, "TACAN Principles and Siting Criteria", Naval Electronic Systems Test and Evaluation Facility, Maryland, 1968.

4. J. H. Larkin, J. McDermott, D. P. Simon and H. A. Simon, "Expert and Novice Performance in Solving Physics Problems", *Science 208*, pp. 1335–1342, 1980.

5. P. E. Slatter, *Building Expert Systems: Cognitive Emulation*, Ellis Horwood, Chichester England, 1987.

6. A. Carlisle Scott, Jan E. Clayton and Elizabeth L. Gibson, *A Practical Guide to Knowledge Acquisition*, Addison-Wesley Publishing Company, 1991.

7. Kay James, "A study of Knowledge Acquisition Methods for the development of an Intelligent Advisor System to assist users in the selection of Tactical Air Navigation System (TACAN) Sites", Report delivered to NRC, March 1992.

Y02329

An Object-Oriented Knowledge Based System for the Evaluation of the SCC Intensity of High Strength Aluminum Alloys

P.R. Roberge
Department of Chemistry and Chemical Engineering
Royal Military College, Kingston, Ontario, Canada, K7K 5L0

#149735

Abstract

Using the structure of a modern programming environment to organize the complex world of materials degradation and lifetime prediction offers two obvious advantages to a knowledge engineer: the first being the advanced level of development most programming environments can provide as various resources and programming facilities. The second advantage of using a functional environment at the conceptual level of a knowledge base system (KBS) project is the fast prototyping permitted by being close to the tools themselves. But the particular advantage of using an object-oriented programming (OOP) environment to effectuate the knowledge transformation is the compatibility of OOP with natural cognitive operations. Knowledge representation goes much beyond conventional data representation and must rely on facts that often possess a halo of fuzziness. The construction of a stress corrosion cracking (SCC) failure predictor in an OOP environment was a revealing exercise. High strength aluminum alloys of the 7XXX family were used as a test case during the development of an empirical model for the prediction of SCC occurrence. While the focus of the present study was mostly the evaluation of risks caused by the possibility of SCC occurrence, it would be relatively simple to broaden the scope of such project to include other failure modes since these are considered to be 'natural' categories fairly well clustered in small regions of parametric space. The complexity of representing both data and the operations to process these data in a knowledgeable way was reduced tremendously by using OOP tools. This paper will demonstrate that by combining procedures and data into objects using OOP techniques the level of abstraction of programming corrosion expertise can be greatly improved.

Introduction

Environmental cracking (EC) is an important form of corrosion mostly because of its insidious and catastrophic character. EC and particularly stress corrosion cracking (SCC) can affect, without much warning, a wide variety of metals and alloys exposed to specific environments. Mechanical forces, which have normally little effect on the general corrosion of metals, will act in synergy with chemical environments and cause the most sudden cracking-type failures of metal structures. The intensity of SCC [1] is affected by a multitude of metallurgical and environmental factors. Their interaction and the mechanisms which underlay their initiation and propagation have been the subject of a multitude of international gatherings. In order to prioritize the parameters involved in SCC or other forms of EC it is important to first rationalize the concepts that have been put forward to describe or explain such types of failure. Some fundamental definitions are required before looking at the specific characteristics of high strength aluminum materials.

Stress Corrosion Cracking Intensity

A general framework to evaluate the thoroughness of given models of SCC problems has been described to consist of six factors. This framework and their factors constitute a description of the independent variables that control the intensity of SCC. The most direct implication of such a dependence or interaction between parameters is that the strength of materials is totally situation dependent and predictable given the proper information [1]. The use of the concept of intensity in the present context is really the first step toward a quantification of the lifetime of a system exposed to a favorable SCC environment. The actual use of the independent variables can be made through correlation equations describing multidimensional operational envelopes or by their incorporation into a knowledge engineering scheme which would also require or include accepted expert advice.

While transgranular SCC is principally controlled by composition and structure, intergranular SCC can proceed in a multitude of ways which depend on the different atomic processes acting concurrently. The principal features governing the **material factor** are the following [1]:

- bulk chemical composition
- crystal structure
- grain boundary composition
- surface condition

The principal features underlying the **environment factor** of situation dependent strength (SDS) have been said to consist of a long list of elements describing the chemical make-up of the environment as well as the aggravating elements or contributors that can be part of operating conditions (wetting and drying, wear and fretting, presence of deposits etc.).

The key features behind the **stress factor** can also be relatively complex and interrelated [1]. But it is important to recognize that two concepts are necessary to describe the global stress factor. The first is the stress definition itself with all its components and the other is the origin of stress or stresses which can vary from external and applied to the wedge action of corrosion products and internal.

Some of the important features controlling the **geometry factor** are difficult to include directly in the calculation of topological dependencies. The following list [1] of such features contains elements that are prime conditions leading to SCC occurrence:

- discontinuities which intensify stress
- generation of galvanic potentials
- chemical crevices
- settling of solids
- restricted geometries leading to concentration cells

These are normally easier to avoid by designing properly but the possibility of these features to occur in a given system and thus the probability of finding SCC problems in its lifetime can serve to estimate the intensity of SCC.

The effect of **temperature factor** on SDS and on the precedent factors can be of different nature. For example the temperature dependence of creep strain at a crack tip differs from the effect temperature has on chemical and electrochemical reaction rates. The temperature of interest to evaluate the intensity of SCC is of course the temperature of the metal surface which is submitted to the forces of all precedent factors. This consideration is particularly important in the case of heat transfer surfaces where the operating conditions can be quite inhomogeneous.

The **time factor** can be considered as an independent variable and, in this respect, is synonymous to the history of a system. The slowly changing conditions of a localized environment with time is a good example of the time factor influence once a system is in service. In such an example a very benign environment can be irreversibly transformed through autocatalytic processes into an absolutely very corrosive one.

Modeling SCC with Object-Oriented Programming

Over the years software developers and researchers have used many different languages to come up with a better way of modeling real-world problems. Because these problems can be quite complex it helps to have methods for reducing the complexity of representing both data and the operations that can be used to process the data. This is what can be achieved best with object-oriented programming (OOP). OOP has been getting much attention lately. In fact, many programming language experts feel that in the 1990s OOP is going to have an impact similar to structured programming in the 1970s [2]. The high-level nature of objects provides numerous benefits. By combining procedures and data into objects the level of abstraction of programming tools can be greatly improved. But the real power of objects is in the way they can be used to define other objects, a process called inheritance.

The object-oriented approach to programming is based on the concepts of encapsulation and extensibility. OOP encapsulates in objects some data and programs to operate on the data. The notion of encapsulation has proven to be a natural and easy paradigm for various application environments such as graphical user-interface systems. The extensibility concept refers to the ability to extend an existing system without introducing changes to it. OOP offers extensibility in two ways [3]: behavior extension and inheritance. The behavior of an object may be extended by including additional programs into it. The inheritance extension, on the other hand, allows the behavior and even the attributes of an object to be reused in the definition of more specialized objects.

The development of an empirical model to evaluate the susceptibility of high strength aluminum alloys to SCC has already been the subject of a previous effort [4] . This project had been realized with an open ended design approach to programming which included the usage of a commercial hypertext system [5]. While such an environment simplified the construction of the user interface or the front end of the system, its maintenance was complicated greatly by the need to switch between different environments. The hypertext environments investigated [6] all suffered from poor connectivity with the traditional algorithms still required to perform calculations. In comparison an OOP information system can provide extensive use of graphic objects in the form of Windows™, a technique readily accepted by the user community.

Amidst the object oriented languages that permit OOP, C++ has recently gained a tremendous momentum with the availability of fast compilers and extensive library of classes and objects. C++, which is a hybrid system based on the original C programming language, is currently the dominant development language in the portable computer (PC) and workstation market [7]. Notwithstanding the numerous qualities and advantages offered by such a programming environment, the high level of programming skills and effort required to build the essential facilities in C++ are serious limitations for the development of demonstration prototypes or the construction and maintenance of elicitation shells.

Fortunately, new software systems are presently sprouting in the field of OOP. The present choice for the implementation of a lifetime prediction elicitation shell is version 1.0 of Paradox™ for Windows™, an environment which combines the facilities of an efficient relational database system with the possibility of programming objects and all the graphical and data exchange interconnectivity provided by the Windows environment.

SCC Intensity of High Strength Al Alloys

The point of view proposed by Staehle [1] is that all engineering materials are reactive chemicals and that the strength of materials depends totally upon the extent to which environments influence the reactivity and subsequent degradation of engineering materials.

Material Factor

The dominant theme for defining the reactivity of materials is the composition of the grain and the grain boundary (GB). The composition of the grain boundary can be either dominated by the formation and concentration of precipitates or by the adsorption and concentration of species collected from the environment. In the case of aluminum alloys, EC fracture is almost exclusively intergranular (IG) unless the mechanical loading conditions are severe and/or the local GBs are unfavorably oriented [8]. The influence of quench rate upon IG and SCC intensity for 2XXX series alloys is relatively well understood. For alloys such as 2024-T3, SCC intensity becomes marked when quench rates fall below 500 C/s and GB regions become sufficiently anodic with respect to grain interiors resulting from GB precipitation of Al_2CuMg. If SCC occurs via a purely anodic dissolution mechanism, as is currently generally accepted for 2XXX series alloys, it may be possible to evaluate SCC intensity by integrating the effects of precipitation at intermediate temperatures during a quench.

For 7XXX series alloys, the picture is less clear-cut. For these alloys, it is generally believed that the influence of quench rate upon SCC is dependent upon an alloy's copper content. For copper-lean alloys, SCC intensity is progressively relaxed by reducing quench rates whereas the reverse is true for alloys containing more than about 1 wt% copper. For most commercial applications of high-strength aluminum alloys, precipitation hardening is of paramount importance. In general, the potential risk of stress corrosion prevents exploitation of the maximum strength offered, because although alloys are immune to SCC in the as-quench condition if GB precipitation is prevented, their SCC intensity increases with precipitation hardening, reaching a maximum before peak strength.

Environment Factor

The large difference in the scale of effect of the environment and the response of a material is important for the organization of testing experiments. Thus, an experiment which would investigate only the nominal condition without consideration for effects such as flow, pH cells, deposits, and other galvanic effects is useless for lifetime prediction. An important point of view implicit in the environment factor of SDS is the difference between nominal and local or near surface environments. Generally, components are designed to resist the nominal environments specified by their applications and planning of testing programs is based on such specifications. It has been shown by Speidel [9] that measured crack velocities can differ by over nine orders of magnitude for a single alloy due to a change in the environment. Aluminum will not corrode without the presence of water but environments as low as 0.8% relative humidity will promote SCC. The chemicals that are the most favorable to crack growth in aluminum are Cl^-, I^-, and Br^-. While other chemicals can have slight effects, these are the most pronounced (with the exception of liquid metal embrittlement by Hg). Cl^- is particularly important since it is present in many environments.

Stress Factor

In the early years of considering SCC, it was common to distinguish between SCC and corrosion fatigue. It is now apparent that there is a continuum between SCC, fatigue corrosion and general fracture (Fig. 1), although there is still no obvious analytical relationship between these phenomena. The effect of stress on SCC is explained in terms of the crack velocity(V) versus linear elastic stress intensity factor (K) diagrams and in terms of time to failure versus applied stress, usually as a function of yield strength. At the stress level where SCC crack growth initiates, the stress intensity factor is given the designation K_{ISCC}. The value of K_{ISCC} is determined in these diagrams by extrapolating to the abscissa at the point where the crack growth rate falls below a specified value.

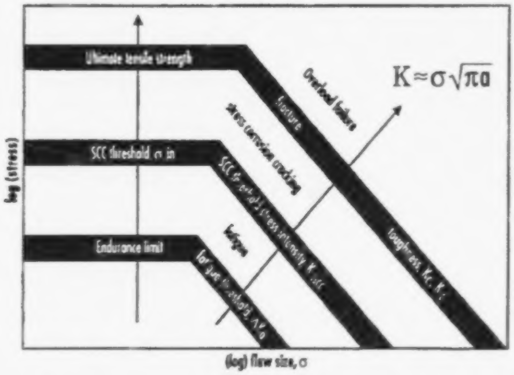

Ultimate tensile strength
SCC threshold σ in
Endurance limit
Overload failure
fracture
stress corrosion cracking
fatigue
$K \approx \sigma\sqrt{\pi a}$
toughness K_C, K_I
SCC threshold stress intensity K_{ISCC}
fatigue threshold ΔK_0
log (stress)
(log) flaw size, a

FIGURE 1: Relative failure boundaries between dry fracture, SCC and fatigue corrosion.

Above K_{ISCC} there are three distinct regions. In the first region (mode I) crack growth rate is dependent on the environment and the level of applied stress. The crack growth rate increases sharply with increased stresses. In the second region (mode II) the rate of crack growth is steady and is independent of the applied stress. This region is controlled by the rate of reaction at the crack tip. In the third region (mode III) the K_I approaches K_{IC} where failure due to yielding of the material occurs. This region is normally small or non-existent. The time to failure vs. applied stress diagrams are used for empirical determination of the design life expectancy and K_{ISCC}. The applied stress can be expressed as the linear elastic stress intensity factor K, or as a percentage of the yield strength. This method is considered more accurate than the V-K diagrams [10].

Geometry Factor

It should be noted that the geometry factor can promote greatly the localization of the environments. For example, tight geometries can produce occluded regions where the environmental chemistry may differ drastically from the nominal bulk environment. Time to failure charts for aluminum alloys show that there is a sharp drop in the time to failure when the pH of the bulk solution is below 11. This does not agree with the Pourbaix diagram for aluminum which shows immunity to corrosion between pH 4 and pH 9. This is attributable to crevice effects that produce a pH of the crack tip significantly lower than the bulk solution. It has been shown that the crack tip can have a pH of approximately 3.5 [11-13]. At this pH, in the normal potential range for aluminum in aqueous environments (-0.4V to -0.9V SHE), the Pourbaix diagram shows a corroding region.

Temperature Factor

An increase in temperature normally decreases the time to failure due to SCC. The effect of temperature on the crack growth velocity is to lower the stress intensity required to initiate SCC and to increase the crack growth velocity. The combined effect is to cause SCC to occur faster at lower stresses. At temperatures below zero the rate of SCC decreases due to the slowing of chemical kinetics and the lack of liquid water. In the particular case of high strength aluminum alloys, temperature excursions in the thermal aging range will affect the strength of the alloys and their intensity of SCC differently depending on their position relative to the peak aged condition and their sensitivity to aging. For 7XXX series alloys the beneficial effects that can result from overaging are strongly influenced by the presence of copper. Alloys with copper contents environments specified by their applications and planning of testing programs is based on such specifications. It has been shown by Speidel [9] that measured crack velocities can differ by over nine orders of magnitude for a single alloy due to a change in the environment. Aluminum will not corrode without the presence of water but environments as low as 0.8% relative humidity will promote SCC. The chemicals that are the most favorable to crack growth in aluminum are Cl^-, I^-, and Br^-. While other chemicals can have slight effects, these are the most pronounced (with the exception of liquid metal embrittlement by Hg). Cl^- is particularly important since it is present in many environments.

Stress Factor

In the early years of considering SCC, it was common to distinguish between SCC and corrosion fatigue. It is now apparent that there is a continuum between SCC, fatigue corrosion and general fracture (Fig. 1), although there is still no obvious analytical relationship between these phenomena. The effect of stress on SCC is explained in terms of the crack velocity(V) versus linear elastic stress intensity factor (K) diagrams and in terms of time to failure versus applied stress, usually as a function of yield strength. At the stress level where SCC crack growth initiates, the stress intensity factor is given the designation K_{ISCC}. The value of K_{ISCC} is determined in these diagrams by extrapolating to the abscissa at the point where the crack growth rate falls below a specified value.

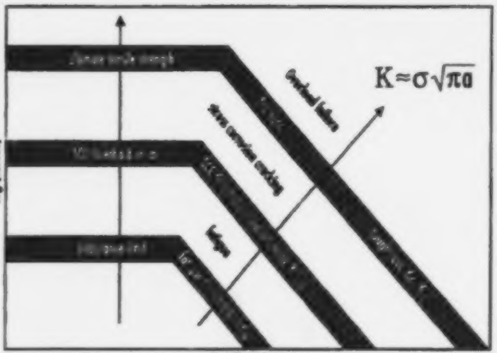


FIGURE 1: Relative failure boundaries between dry fracture, SCC and fatigue corrosion.

Above K_{ISCC} there are three distinct regions. In the first region (mode I) crack growth rate is dependent on the environment and the level of applied stress. The crack growth rate increases sharply with increased stresses. In the second region (mode II) the rate of crack growth is steady and is independent of the applied stress. This region is controlled by the rate of reaction at the crack tip. In the third region (mode III) the K_I approaches K_{IC} where failure due to yielding of the material occurs. This region is normally small or non-existent. The time to failure vs. applied stress diagrams are used for empirical determination of the design life expectancy and K_{ISCC}. The applied stress can be expressed as the linear elastic stress intensity factor K, or as a percentage of the yield strength. This method is considered more accurate than the V-K diagrams [10].

Geometry Factor

It should be noted that the geometry factor can promote greatly the localization of the environments. For example, tight geometries can produce occluded regions where the environmental chemistry may differ drastically from the nominal bulk environment. Time to failure charts for aluminum alloys show that there is a sharp drop in the time to failure when the pH of the bulk solution is below 11. This does not agree with the Pourbaix diagram for aluminum which shows immunity to corrosion between pH 4 and pH 9. This is attributable to crevice effects that produce a pH of the crack tip significantly lower than the bulk solution. It has been shown that the crack tip can have a pH of approximately 3.5 [11-13]. At this pH, in the normal potential range for aluminum in aqueous environments (-0.4V to -0.9V SHE), the Pourbaix diagram shows a corroding region.

Temperature Factor

An increase in temperature normally decreases the time to failure due to SCC. The effect of temperature on the crack growth velocity is to lower the stress intensity required to initiate SCC and to increase the crack growth velocity. The combined effect is to cause SCC to occur faster at lower stresses. At temperatures below zero the rate of SCC decreases due to the slowing of chemical kinetics and the lack of liquid water. In the particular case of high strength aluminum alloys, temperature excursions in the thermal aging range will affect the strength of the alloys and their intensity of SCC differently depending on their position relative to the peak aged condition and their sensitivity to aging. For 7XXX series alloys the beneficial effects that can result from overaging are strongly influenced by the presence of copper. Alloys with copper contents below around 1 wt% require gross overaging before high resistance to SCC can be obtained, whereas alloys with higher copper contents need only sacrifice 10 to 15% of the maximum strength to minimize service problems.

Time Factor

Time is an important and non trivial factor for the construction of a predictive model. Failures have occurred in a broad time period ranging from prior to installation to 30 years in service. The identification of the mechanism of failure may be confirmed if the SCC lifetime estimate of the component was comparable to the actual failure time. Wood and Gerberich [14] reviewed five models in the literature for 7XXX alloys which ranged from purely theoretical models to semi-empirical and fully empirical models. The majority of the models consisted of curve fitting to the data. Wood and Gerberich concluded that none of the proposed models could be used for prediction of SCC as their theoretical basis and accuracy were suspect. In a more recent paper Onoro et al. [15] presented a semi-empirical model for 7079 aluminum alloys that used 10 constants related to the material, its heat treatment and the environment. While the data presented in [15] was accurate at predicting the V-K diagram of the 7079 alloy developed from double cantilever beam specimens no attempt was made to compare it to actual service failures.

Failure Predictor

Uncertainty and Inexact Knowledge

Classical probability theory expresses the probability of an event as the frequency of occurrence of that event in a given set of conditions. This is also called *a priori* probability because the outcome of an event is already known [16]. This theory assumes that all possible events are known and that each is as likely to occur as any other. When the probability of two events are not independent of each other this is called conditional probability. Conditional probability calculates the probability of event A given that event B occurred. In failure analysis the opposite is more important i.e. given that A occurred what is the probability that B (the earlier event) occurred? This is called inverse probability, or when the rules or personal experiences subjectively link effects to causes, 'degree of belief' [17] as it represents an estimate of the truth but is not itself verifiable. There are basically three ways, of increasing complexity, to represent uncertain information in knowledge engineering: subjective probabilities, certainty factors (CF) and fuzzy logic [18]. The CF approach gives a good approximation of the measure of belief without being too complex to manage during the knowledge elicitation. The concept of CF was developed for the Mycin project [19] and has been used successfully in many other systems [20]. In this context, a CF is a numerical value that expresses the extent to which a given conclusion is acceptable on the basis of a given set of evidence. A CF with a value of 1 indicates total belief, whereas a CF with a value of -1 indicates total disbelief [21].

The two main principles for applying CF to a particular knowledge engineering situations are: 1) the CF is a quantity that describes the credibility of a given conclusion and 2) the rules must be structured in such a way that any particular rule will either add to the belief in a given conclusion or add to its disbelief. In Eq. 1, which summarizes this second principle, MB is a measure of belief of event P occurring given the occurrence of event E and MD is a measure of disbelief.

$$CF(P,E) = MB(P,E) - MD(P,E) \quad (1)$$

According to the second principle, if $MB(P,E) > 0$ then $MD(P,E) = 0$ and if $MD(P,E) > 0$ then $MB(P,E) = 0$. By definition MB and MD need to be calculated for each possible conclusion of a system. This calculation is performed by first initializing both terms to zero and then incrementally including the effect of each additional fact or rule. The process of unfolding an increasing or decreasing value of CF is at the same time both instructive and flexible since the rules do not have to be considered in any specific order but their impact is immediately reflected in the modification of CF. The measure of belief that results from considering two sources of evidence, rule 1 (R1) and rule 2 (R2), can be calculated by using Eq. 2 when these sources have MBs > 0 or with Eq. 3 for MDs > 0.

$$CF(P,E) = MB(R_1) + MB(R_2)\,(1 - MB(R_1)) \quad (2)$$

$$CF(P,E) = MD(R_1) + MD(R_2)\,(1 - MD(R_1)) \quad (3)$$

Failure Predictor 1

Human experts sort evidences by experience. Based on their assessment of a given set of information, experts will form an initial hypothesis and determine what additional information or tests are required to prove or disprove this initial hypothesis. Further information normally raises more questions and experts may go through several iterations before feeling confident that the causes and mechanisms of a failure are really determined with a certain degree of confidence. The first prototype of Failure Predictor was also structured from the general to the specific. The first operation of the program, level one, was a general evaluation of the operating conditions to determine which mechanisms could operate under the conditions given. Then the user would select a specific mechanism to consider. During the next two levels an increasingly more detailed evaluation of the mechanisms was done. Changes in CF values observed during the evaluation process were used to estimate if the program was approaching or receding from the objective. At the third level the user was finally asked questions concerning the microstructure of the material.

Failure Predictor 1 was strongly dependent on the relative value and weight of the various factors used to estimate probabilities of fracture. The contribution of humidity, temperature and other parameters had to be quantified before their use by the program. The overall outcome of SCC was dependent on three factors, so each factor was broken down into its sub-factors. The scale chosen was ±100 as some evidence (such as very low temperature) decreases the probability of SCC. Each sub-factor was then assigned a value between -100 and +100 based on its contribution to SCC.

The validation of Failure Predictor was done with series of test cases taken from the literature and the results obtained with Failure Predictor were compared either to the conclusions reported for these cases or to diagnoses given by three human experts. Failure Predictor fully passed the Turing test which states that a user presented with the results from a human expert and the results from an expert system (ES) should not be able to tell the difference.

Failure Predictor 2

While the second Failure Predictor prototype gets its information from the knowledge base (KB) developed for the first prototype, its organization and operation are entirely different. Failure predictor 2 was designed with an architecture integrating the specialized knowledge of SCC, described earlier in the paper, and the organization of objects in OOP. Inheritance is a natural quality which as fascinated mankind since the beginning of time. Nature took billions of years to generate life in all its present forms and the summation of that effort is passed on to every descendant of each species each time the reproduction process is repeated. The development of programming languages that could mimic this powerful property of Nature led eventually to OOP and C++. Inheritance is probably the single feature in C++ that provides the most power to the class concept. From a programming point of view, inheritance makes it possible to derive one class from another and thus give to a program a high level of flexibility while keeping its development cost low. From a conceptual point of view, inheritance forces a system developer to think in terms of appurtenance between objects instead of thinking in terms of mathematical relations as it has to be done with structured programming. Whether and when one class is derived from another is purely the developer's decision but that decision must be made at compile time and cannot be altered during runtime. Such an early decision requirement is considered to be a limitation by programmers. But for projects such as Failure Predictor the concept of inheritance and its implementation in objects offers a unique opportunity. It makes possible the construction of programs closely fitting the knowledge premises while keeping the interaction between the parameters very flexible.

From a materials science and materials engineering point of view, the possibility of transforming knowledge into programmable objects and classes offers the important advantage of being compatible with the type of information which has to be processed. Figures 2 and 3 illustrate how the knowledge of SCC would be organized in an OOP environment such as C++ or Paradox for Windows. The specific interest of a user of Failure Predictor 2 would be satisfied by bringing the focus on the domain of interest. In Fig. 2, for example, the focus is on the material factor of SCC intensity and on its sub components, while the role of the other factors is more discrete. Any modification to the sub components of the material factor would be directly reflected in alterations to the SCC intensity itself and to any other parameter directly or indirectly involved.

The construction of any other factor, such as the geometry factor presented in Fig. 3, would be simply a duplication of the first module with minor alterations to reflect the differences. In fact, it becomes easy to imagine how such a system could be made either more global or, at the opposite, could be made more specialized and complicated. The SCC intensity could thus become itself a component of a larger Materials Degradation expertise shell or, in turn, each sub component of the six factors of SCC intensity could become the focus of interrogation and be broken into their own sub components.. The grain boundary composition sub component of the material factor, for example, could be expanded into its own factors which would reflect the specific knowledge on either an alloy or a class of alloy.

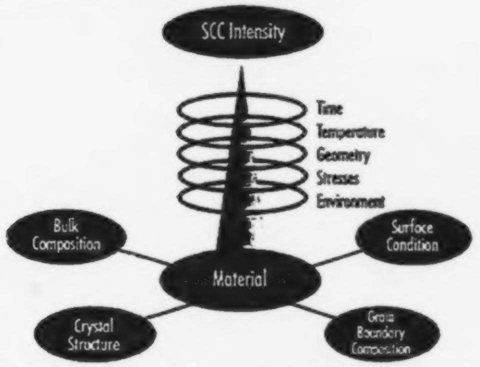


FIGURE 2 - An OOP representation of the material factor of SCC intensity.

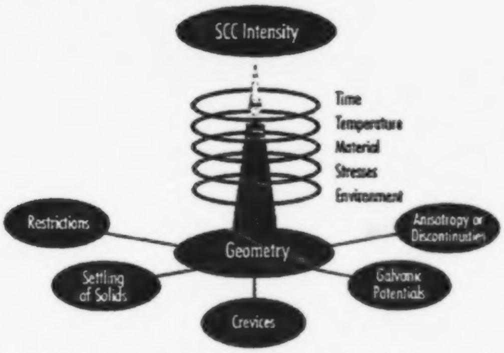


FIGURE 3 - An OOP representation of the geometry factor of SCC intensity.

By going deeper in the knowledge which is being organized for computerization, the construction of a tool such as Failure Predictor brings to the forefront some of the advantages which have been associated with the development of ESs, i.e. areas of insufficient knowledge are revealed, the structure of the tool becomes an intelligent checklist and the tool can be used as an expert sharpener.

Conclusion

While the focus of the present study was mostly the evaluation of risks caused by the possibility of SCC occurrence, it would be relatively simple to broaden the scope of such project to include other failure modes since these are considered to be 'natural' categories fairly well clustered in small regions of parametric space [22]. Using a programming environment to organize the complex world of materials degradation and lifetime prediction offers the obvious advantage that most modern programming environments can give, i.e. providing resources and facilities to support and assist programmers. But the particular advantage of using an OOP environment to effectuate the knowledge transformation is the compatibility of OOP with natural cognitive operations such as classifying, instantiating and measuring. Knowledge representation goes much beyond conventional data representation and must rely on facts that often possess a halo of fuzziness. The construction of Failure Predictor in an OOP environment was a revealing exercise since it not only set up a framework that can help a person to learn the subtleties that compound the SCC intensity of high strength aluminum alloys but it also permitted to start thinking in terms of failure and lifetime prediction that could lead to the creation of a much broader expertise shell.

References

1. R.W. Staehle, "Understanding Situation-Dependent Strength: a Fundamental Objective in Assessing the History of Stress Corrosion Cracking", Proc. Environment-Induced Cracking of Metals, NACE, Houston, TX, 1989, 561-612.
2. K. Weiskamp and B. Flamig, *The Complete C++ Primer*, Academic Press, N.Y., U.S.A., 1990, 28.
3. W. Kim, *Introduction to Object-Oriented Databases*, The MIT Press, 1990, 9.
4. G.D. Bryson and P.R. Roberge, "Development of an empirical model for the Evaluation of Susceptibility to SCC", Proc. Materials Performance: Sulphur and Energy, CIM, Montreal, QC, 1992, 247-257.
5. G.D. Bryson, "Material Analyst: an Expert System for the prediction of Material Maintenance", Master thesis, Royal Military College of Canada (RMC), Kingston, ON, 1992.
6. P.R. Roberge, "The Hypermedia Technology as a Front End to Knowledge Processing Systems", Proceedings of the Fourth Symposium/Workshop

intensity could become the focus of interrogation and be broken into their own sub components.. The grain boundary composition sub component of the material factor, for example, could be expanded into its own factors which would reflect the specific knowledge on either an alloy or a class of alloy.

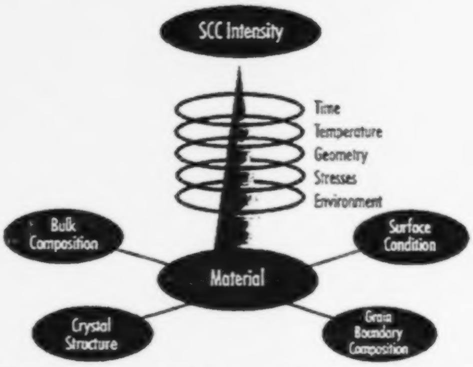


FIGURE 2 - An OOP representation of the material factor of SCC intensity.

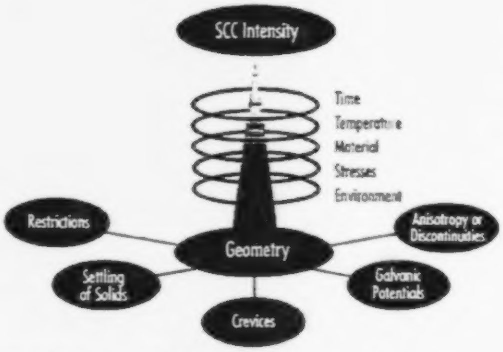


FIGURE 3 - An OOP representation of the geometry factor of SCC intensity.

By going deeper in the knowledge which is being organized for computerization, the construction of a tool such as Failure Predictor brings to the forefront some of the advantages which have been associated with the development of ESs, i.e. areas of insufficient knowledge are revealed, the structure of the tool becomes an intelligent checklist and the tool can be used as an expert sharpener.

Conclusion

While the focus of the present study was mostly the evaluation of risks caused by the possibility of SCC occurrence, it would be relatively simple to broaden the scope of such project to include other failure modes since these are considered to be 'natural' categories fairly well clustered in small regions of parametric space [22]. Using a programming environment to organize the complex world of materials degradation and lifetime prediction offers the obvious advantage that most modern programming environments can give, i.e. providing resources and facilities to support and assist programmers. But the particular advantage of using an OOP environment to effectuate the knowledge transformation is the compatibility of OOP with natural cognitive operations such as classifying, instantiating and measuring. Knowledge representation goes much beyond conventional data representation and must rely on facts that often possess a halo of fuzziness. The construction of Failure Predictor in an OOP environment was a revealing exercise since it not only set up a framework that can help a person to learn the subtleties that compound the SCC intensity of high strength aluminum alloys but it also permitted to start thinking in terms of failure and lifetime prediction that could lead to the creation of a much broader expertise shell.

References

1. R.W. Staehle, "Understanding Situation-Dependent Strength: a Fundamental Objective in Assessing the History of Stress Corrosion Cracking", Proc. Environment-Induced Cracking of Metals, NACE, Houston, TX, 1989, 561-612.
2. K. Weiskamp and B. Flamig, *The Complete C++ Primer*, Academic Press, N.Y., U.S.A., 1990, 28.
3. W. Kim, *Introduction to Object-Oriented Databases*, The MIT Press, 1990, 9.
4. G.D. Bryson and P.R. Roberge, "Development of an empirical model for the Evaluation of Susceptibility to SCC", Proc. Materials Performance: Sulphur and Energy, CIM, Montreal, QC, 1992, 247-257.
5. G.D. Bryson, "Material Analyst: an Expert System for the prediction of Material Maintenance", Master thesis, Royal Military College of Canada (RMC), Kingston, ON, 1992.
6. P.R. Roberge, "The Hypermedia Technology as a Front End to Knowledge Processing Systems", Proceedings of the Fourth Symposium/Workshop

on Applications of Expert Systems in DND, RMC, Kingston, ON, 1992, 153-168.

7. D. Taylor, *Object Oriented Information Systems*, John Wiley and Sons, 1992, 160.

8. N.J.H. Holroyd, "Environment-Induced Cracking of High-Strength Aluminum Alloys", Proc. Environment-Induced Cracking of Metals, NACE, Houston, TX, 1989, 311-345.

9. M.O. Speidel. "Interaction of Dislocations with Precipitation in High Strength Aluminum Alloys and Susceptibility to SCC", Fundamental Aspects of Stress Corrosion Cracking, NACE, Houston, TX, USA, 1969, 561.

10. Mills, K., *Failure Analysis and Prevention. - Metals Handbook*, 9th Edition, Vol. 11, ASM International, Metals Park Ohio, USA, 1986, 53.

11. M.G. Fontana and R.W. Staehle, *Advances in Corrosion Technology*, Vol. 2. Plenum Press, New York, 1972, 246.

12. M.P. Mueller, A.W. Thompson and I.M. Bernstein, "Stress Corrosion Behaviour of 7075 Aluminum in 1N Aluminum Chloride Solutions" . *Corrosion*, Vol. 41, No 3, 1985, 127-136.

13. R.T. Foley and H.L. Anh, "On the Nature of the Occluded Cell in the Stress Corrosion Cracking of AA 7075-T651 - Analysis of the Solution Inside the Crack", *Corrosion*, Vol. 40, No 4, 1984, 195-197.

14. W.E. Wood and W.W. Gerberich, "The Mechanical Nature of Stress-Corrosion Cracking in Al-Zn-Mg Alloys: 2. Electrochemical-Mechanical Model", *Met. Trans.*, Vol. 5. 1974, 1295-1304.

15. J. Onoro, A. Moreno and C. Ranninger, "Stress Corrosion Cracking Model in 7075 Aluminum Alloy", *J. Mater. Sci.*, Vol. 24, 1984, 3888-3891.

16. J. Giarratano and G. Riley, *Expert Systems Principles and Programming*, PWS-KENT Publishing Company, Boston, 1989, 190.

17. S.L. Tanimoto, *Elements of Artificial Intelligence - An Introduction Using LISP*, Computer Science Press, Maryland, 1987, 240.

18. E. Turban, *Decision Support and Expert Systems: Management Support Systems*, 2nd Edition. Macmillan, New York, 1990, 522.

19. B. Buchanan and B. Shortliffe, *Rule-Based Expert Systems*, Addison-Wesley, Menlo Park, CA, 1984.

20. B. Buchanan R. and Duda, "Principles of Rule-Based Expert Systems," in *Ad. Comp.*, Vol. 22, M. Yovits (Ed.), Academic Press, New York, NY, 1983.

21. D.W. Rolston, *Principles of Artificial Intelligence and Expert Systems Development*, McGraw-Hill, New Yorc, NY, 1988

22. J.P. Morrill and D. Wright, *Trans. ASME*, Vol. 111, No 3, 1989, 306-310.

Vers une Architecture d'Agent pour le Commandement et Contrôle (C^2)

B. Chaib-draa, P. Levesque. E. Paquet
Département d'Informatique, Université Laval
Sainte-Foy, Québec, G1K 7P4 Canada
email: chaib@ift.ulaval.ca
fax: (418) 656-2324

#149738

L. Lamontagne
Centre de Recherche de Valcartier–CRDV
P.O. Box 8800, Courcelette, Québec, G0A 1R0 Canada

August 5, 1993

Résumé– *Cet article présente un modèle d'architecture permettant de comprendre, d'expliquer et de concevoir des systèmes adaptatifs d'agents, capables de coordonner leurs activités. Plus précisément, l'architecture proposée combine les avantages des systèmes réactifs, des systèmes de planification et des systèmes de prise de décision. De cette manière, tout agent a alors la capacité de faire face aux situations routinières, familières et non-familières. Cette architecture est ensuite appliquée à la modélisation du commandement et contrôle (C^2) considéré comme un processus. Des détails et des explications sont fournis dans ce cadre, sur la manière dont un commandement donné transmet ses ordres à ses subordonnés et coordonne leurs activités. Finalement, l'implantation et la validation du modèle sont présentées.*

1 Introduction

Les modèles de prise de décision pour environnements distribués tels qu'ils existent actuellement s'avèrent limités, particulièrement pour la modélisation et l'analyse en commandement et contrôle (C^2). Ce type d'environnement fort complexe ne peut être modélisé et analysé en utilisant uniquement les outils mathématiques classiques de prise de décision. En effet, ce type d'environnement fait généralement intervenir plusieurs entités hétérogènes capables de coopérer, de coexister ou de rentrer en compétition. Chacune de ces entités a ses propres connaissances, ses propres capacités et ses propres buts, et peut à tout moment agir sur la base de décisions faisant intervenir ce que veut cette entité et ce que veulent les autres entités de son environnement. C'est pourquoi il paraît de plus en plus necessaire d'étudier ce genre d'environnement en utilisant diverses méthodes issues de diverses disciplines: mathématiques, linguistique, sciences cognitives, intelligence artificielle, théorie de la décision, informatique répartie, etc. Ce genre d'étude peut aujourd'hui être mené dans le cadre des environnements multiagents et plus généralement dans le cadre de la modélisation et la programmation des systèmes intelligents distribués.

Les environnements multiagents visent à concevoir des systèmes permettant de modéliser le comportement d'un ensemble d'entités plus ou moins expertes, plus ou moins organisées selon des lois de type social. Ces entités ou "agents" disposent d'une certaine autonomie et sont immergées dans un environnement dans lequel et avec lequel elles interagissent. Dans ce contexte, le centre de recherche de la défense de Valcartier (CRDV) a tenu à évaluer l'apport des environnements multiagents au domaine de l'aide à la décision en commandement et contrôle. Pour cela, le CRDV a décidé d'initier une recherche devant mener à la conception et à l'implantation d'une architecture logicielle d'un système intelligent capable d'agir et d'interagir dans des environnements multiagents.

Dans une première phase de cette recherche, une étude exploratoire basée sur une revue de l'existant a été menée. Cette étude exploratoire nous a alors mené vers la conception et la réalisation d'une architecture d'agent capable d'avoir un comportement conforme aux situations de décisions, habituellement observées dans tout environnement multiagent: les routines, les situations familières et les situations non-familières (ou situations de crise). Ainsi, face à toute situation routinière, l'architecture proposée se comporte comme un système réactif, tandis que face à une situation familière, elle se comporte comme un système de planification. Finalement, face à toute situation non-familière, l'architecture proposée se comporte comme un système de prise de décision.

Un cas particulier important de l'application de l'architecture est la modélisation de C^2 considéré comme un processus. A cet effet, la troisième partie du présent article montre les limitations des modèles classiques du type Lawson [4] et l'apport de notre architecture à la modélisation du traitement d'information dans le commandement et contrôle. À cet effet, des détails et des explications sont fournis sur la manière dont un commandement donné transmet ses ordres à ses subordonnés et coordonne leurs activités. Finalement, quelques applications de l'architecture proposée, dans le cadre du commandement et contrôle, sont discutées:

Le présent article fait état de nos travaux en vue de réaliser l'architecture proposée, qui une fois validée, devra être appliquée au commandement et contrôle.

2 Architecture d'agent basée sur les comportements: une vue générale

La conception de notre architecture d'agent a été motivée par le besoin de combiner les avantages complémentaires des systèmes réactifs, des systèmes de planification et des systèmes de prise de décision. L'architecture proposée [2, 3] nécessitait donc, d'être en premier lieu réactive pour être en mesure de répondre rapidement aux changements brusques de l'environnement. Elle devrait, en second lieu, permettre la planification des actes qui réalisent un but (ou une tâche) identifié(e), reconnu(e) ou directement perçu(e) par l'agent lorsqu'il se trouve en situation familière. Finalement, le modèle proposé devrait permettre la prise de décision pour chacun des participants dans le cas où il doit faire face à des situations non-familières ou des situations de crise. Ces considérations nous ont donc amené à proposer l'architecture d'un système intelligent dont le flot de données est schématisé en figure 1. Cette architecture nous a été inspirée par les travaux cognitifs relatifs aux comportements, et plus particulièrement, les travaux de Rasmussen [6] relatifs aux tâches cognitives. Nous reviendrons sur ce choix dans la partie 3.1.

2.1 Comportement de l'agent face à différentes situations de décisions

Le modèle d'agent proposé a été développé à partir de l'analyse du comportement humain et inclue les différentes phases suivantes. Tout d'abord, l'agent perçoit une ou plusieurs informations venant de son environnement. Cette information le pousse soit à agir si elle est directement perçue sous forme d'action, soit à planifier si elle perçue comme une tâche ou un but. Si en revanche, l'information n'est perçue sous aucune de ces deux formes, l'agent doit alors identifier ou reconnaître une situation. L'identification et la reconnaissance lui permettent à leur tour, soit d'agir si la situation est identifiée ou reconnue en termes d'action, soit de planifier si la situation est identifée ou reconnue en termes de but ou de tâche. Enfin, si l'agent est face à une ambiguïté, ou s'il lui est nécessaire d'élaborer (et de choisir) un but parce qu'il se trouve en face d'une situation inhabituelle, alors il doit prendre des décisions qui doivent l'amener à s'engager dans une action ou un but donné.

2.2 Les quatre niveaux de l'architecture

L'architecture proposée comprend quatre niveaux: le niveau réflexe, le niveau planification, le niveau prise de décision et enfin, le niveau raisonnement.

a) le niveau réflexe: dans ce niveau, l'information perçue suit l'un des deux chemins suivants:

i) *perception ⟶ action*

ii) *perception ⟶ identification ou reconnaissance ⟶ action*

L'objectif de ce niveau est de fournir à l'agent des capacités de traitement pour les changements rapides ayant lieu dans son environnement. Généralement, le comportement d'agent intégré à ce niveau est sensé représenter la performance de l'agent durant des *routines*, c'est-à-dire des actes sans contrôle "conscient". A signaler par exemple, que dans un contexte d'aide à la décision, le niveau reflexe permet de résoudre les situations de décision faciles (où l'information en entrée est fiable et les conséquences des actions bien cernées). Dans ce contexte, le module *Perception* indique au module *Action* la décision qui est la plus appropriée, selon les connaissances de l'agent et en fonction des informations reçues. La tâche du module *Action* est alors de transmettre cette information à l'environnement via des effecteurs. Il convient de préciser, qu'a ce niveau l'information est généralement perçue sous forme signal. Un exemple concret est celui d'un véhicule intelligent qui "surveille constamment sa route". A certains moments cependant, l'information peut être perçue sous formes de signes relatifs à certains comportements qu'il convient de déclencher.

b) le niveau planification: ce niveau est typiquement contrôlé de manière "consciente" et l'information perçue suit l'un des chemins suivants:

i) *perception ⟶ planification ⟶ action*

ii) *perception ⟶ identification ou reconnaissance ⟶ planification ⟶ action*

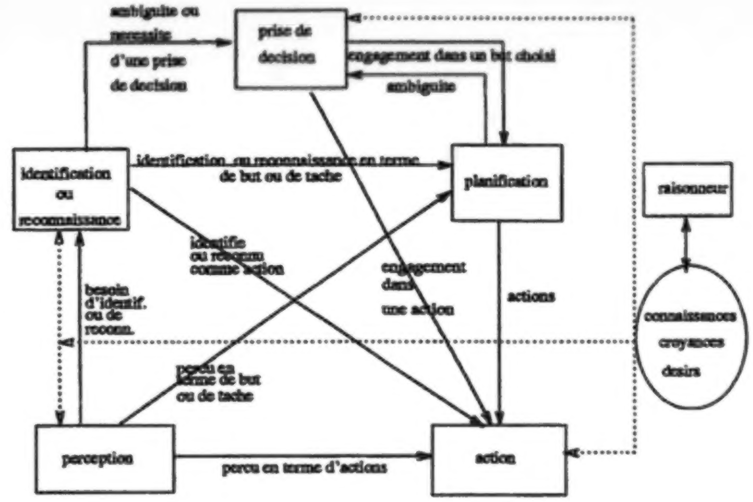

ambiguite ou
necessite
d'une prise
de decision
prise de
decision
engagement dans un but choisi
ambiguite
identification
ou
reconnaissance
identification ou reconnaissance en terme
de but ou de tache
planification
raisonneur
identifie
ou reconnu
comme action
engagement
dans
une action
actions
besoin
d'identif.
ou de
reconn.
connaissances
croyances
desirs
percu en
terme de but
ou de tache
perception
percu en terme d'actions
action

Figure 1: Architecture d'agent basée sur les situations de décisions

Habituellement, ce niveau traite des ***situations familières*** pour lesquelles les solutions sont suggérées par des schémas (patterns) issus de la perception, de l'identification ou de la reconnaissance. À noter que l'information à ce niveau est typiquement perçue sous forme de signes servant à activer ou à modifier des comportements prédéterminés.

c) Le niveau prise de décision: ce niveau est soit typiquement confronté avec l'ambiguïté entre différents choix, soit avec la non possibilité d'identifier et/ou de reconnaître une situation donnée en termes d'action ou de but. Généralement, le flot de données suit le chemin suivant:

perception ⟶ identification ou reconnaissance ⟶ prise de décision ⟶ planification ⟶ action

À ce niveau généralement, l'agent est confronté à des ***situations non-familières*** pour lesquelles il a besoin soit de lever l'ambiguïté s'il a à choisir entre plusieurs options; soit de choisir un but en fonction des intentions des autres agents qui coexistent dans le même environnement que lui. Pour ce niveau, l'information est généralement perçue sous forme symbolique nécessitant un raisonnement formel.

d) le niveau raisonnement: l'agent doit dans ce cas contrôler les autre niveaux et mettre à jour continuellement sa base de connaissances et de croyances. Cette mise à jour est effectuée en fonction des informations perçues et en fonction des informations provenant des autres modules. Le contrôle et la mise à jour sont assurées par le raisonneur qui agit parallèlement à ce qui se passe dans les trois autres niveaux.

Il convient de noter que pour les situations non-familières, l'architecture d'agent proposée utilise un raisonnement basé sur les cartes cognitives [1] de façon à permettre à chaque agent de raisonner sur ses buts et ses croyances tenant compte de ceux des autres agents. De plus, des connaissances *sociales* ont été intégrées à cette architecture de manière à renforcer les niveaux routines et situations famileres, et permettre ainsi à chacun de coordonner ses activités avec d'autres agents dans le cas des situations de ce type. Parmi ces connaissances (connues généralement par tous les agents), il convient de citer:

1. *les normes:* ce sont des prévisions spécifiant les comportements appropriés pour des situations données. Ces prévisions sont partagées par tous les agents.
2. *les règles de coordination:* ce sont des règles sociales comme par exemple conduire à droite, répondre à une question, etc. dont le but est d'assurer la coordination entre agents qui essayent de faire plus ou moins la même chose avec le minimum d'inférence entre eux.
3. *les règles coopératives:* ces règles reflètent des comportements d'agent dont la performance à une action n'a de sens que si les autres agents font la même chose. Tel est le cas par exemple, du recyclage des papiers ou des bouteilles, etc.
4. *les règles collectives:* ce sont des règles promulguées, dictées, ou imposées au groupe d'agents par une autorité quelconque, reconnue mutuellement.

Bien entendu, les méthodes de raisonnement changent en fonction des situations de décisions. C'est pourquoi, il a été décidé d'avoir recours à différents mécanismes de raisonnement:

- un raisonnement procédural, dans le cas des routines, pour lequel les procédures sont représentées par des MOPs (memory organized packages) [7];
- un raisonnement basé sur des règles de production ou sur des cas similaires, pour les situations familières;
- un raisonnement formel basé sur les cartes cognitives, pour les situations non-familières (ou de crise).

L'architecture, dont nous venons de donner un aperçu général, est issue de l'analyse des tâches cognitives, et par conséquent elle fait intervenir les differentes situations de décisions habituellement observées dans les comportements humains. Elle peut donc s'avérer intéressante pour modéliser le processus de commandement et contrôle. C'est ce que s'attache à détailler la section ci-dessous.

3 Apport de l'architecture proposée au commandement et contrôle (C^2)

3.1 Le commandement et contrôle vu comme un processus

Le commandement et contrôle (C^2) est généralement perçu de différentes manières selon qu'on le considère sous l'aspect théorie, conception ou évaluation. C'est ainsi qu'il est considéré soit comme un système soit comme un processus. Selon Metersky [5], le commandement et contrôle (C^2) peut être divisé selon différentes composantes comme le montre la figure 2.

Ainsi au niveau le plus haut, l'aspect C^2-système (i.e. l'aspect système de C^2) est relié aux différents systèmes d'armements de façon à permettre de "mapper" les décisions prises (au niveau C^2-système) aux différents systèmes d'armements considérés. L'avantage d'une telle représentation réside principalement dans le fait que C^2 a la même allure qu'on soit dans la marine ou dans les forces aériennes, pourvu qu'on change justement la composante systèmes d'armements de façon à tenir compte du corps d'armée considéré. La composante C^2-système est une combinaison d'éléments qui forment un tout complexe. Cette composante comprend deux parts interdépendantes: 1) le C^2-processus (i.e. l'aspect processus de C^2) qui consiste en un ensemble d'opérateurs humains, de procédures et d'organisations et 2) les composantes physiques comme par exemple: les senseurs, les ordinateurs, les communications, etc. La fonction majeure de C^2-processus consiste à fournir des décisions spécifiques aux décideurs responsables, tandis que les composantes physiques sont chargées de fournir le mécanisme servant à implanter le C^2-processus.

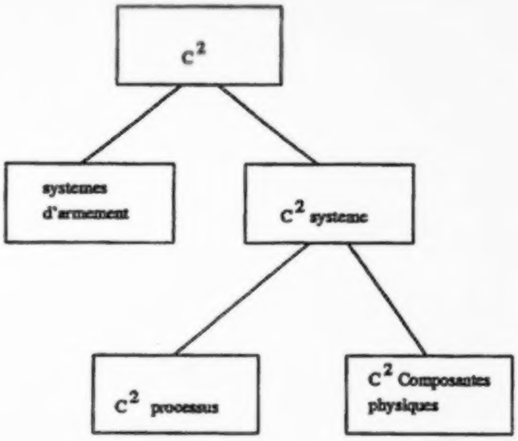


Figure 2: Les différentes composantes du commandement et contrôle

Dés lors C^2-système peut être vu selon trois perspectives: le système total C^2, C^2-processus et les composantes physiques. Cette manière de voir le commandement et contrôle permet de séparer les éléments individuels de C^2 afin de mieux le modéliser et l'évaluer. A cela, s'ajoute le fait que différentes techniques issues de différentes disciplines sont requises à l'étude et à la compréhension des aspects internes et externes de C^2-process, des composantes physiques, ainsi que du système total C^2. C'est ainsi que le génie des systèmes et la théorie du contrôle des systèmes sont les mieux adaptés pour les systèmes physiques. Tandis que l'analyse cognitive du comportement et l'analyse organisationnelle sont les mieux adaptés pour C^2-processus (d'où notre choix pour une architecture basée sur les travaux cognitifs issus des comportements). La théorie du contrôle des systèmes peut être également utilisée pour le système total C^2. Cette théorie a toutefois montré ses limites, particulièrement pour la modélisation entre les différents C^2-processus d'une part, et entre les C^2-processus et les composantes physiques d'autre part.

Dans le reste de cette section, on montrera comment les concepts développés dans le cadre de la présente recherche permet de pallier cela. A cet égard, nous présentons le C^2-processus tel qu'il a été élaboré pour la première fois par Lawson [4] et nous montrerons ses limites. Nous verrons ensuite comment notre architecture d'agent peut améliorer le

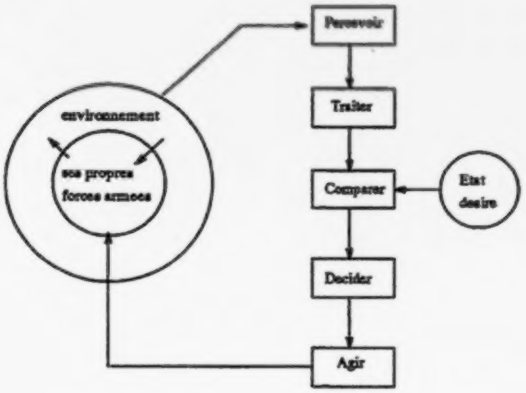


Figure 3: Le modèle C^2-processus de Lawson

C^2-processus tout en permettant la modélisation entre différents C^2-processus et différentes composantes physiques.

3.2 Le modèle C^2-processus de Lawson

Il apparaît clairement, suite aux considérations introduites précédemment, que le composant C^2-processus constitue l'épine dorsale du C^2 global. Ce composant pourrait être étudié en utilisant des concepts avancés issus des sciences cognitives, de l'intelligence artificielle et des sciences organisationnelles. Ce genre d'étude s'avère généralement très utile pour la simulation, la formation, l'aide à la décision, et la coopération personne(s)-machine(s). Avant de voir de plus près comment pourrait s'effectuer ce genre d'apport, il convient tout d'abord de modéliser convenablement l'aspect processus de C^2.

Selon Lawson [4], un C^2-processus peut être vu de la manière suivante: une *haute autorité* spécifie un état désiré de l'environnement (ou d'une zone géographique) qu'il essaye d'atteindre et de maintenir (quitte à employer la force si c'est necessaire) en utilisant le commandement et contrôle (on pourrait voir cela comme une *mission*). Vu sous cet angle, le commandement et contrôle doit incorporer la capacité de: 1) percevoir l'état de son environnement; 2) comparer ce qu'il perçoit avec l'état désiré et 3) agir sur son environnement de manière à ce que l'état du monde correspond à l'état désiré. A cet effet, Lawson propose le modèle présenté en figure 3.

Ce modèle nous paraît limité dans la mesure où il est linéaire et incomplet. La linéarité impose par exemple de suivre toutes les étapes une après l'autre, ce qui a pour inconvénient de ne pas représenter les situations de décisions du type réflexe ou du type familier. En fait, pour Lawson toutes les situations sont traitées de la même manière. Ainsi par exemple, une situation perçue directement sous la forme d'une action à effectuer (réflexe) doit donner lieu, comme toute autre situation, à un traitement et à une prise de décision, ce qui est loin de refléter la réalité. De plus, le modèle de Lawson est incomplet car il ne tient pas compte de la planification, c'est à dire de la manière de réaliser un but donné et des moyens qu'il convient de mettre en oeuvre. D'ailleurs, un module de planification est nécessaire dans C^2 dans la mesure où il pourrait être étendu pour prendre en compte l'*allocation des ressources*.

L'architecture proposée dans le cadre de cet article permet de pallier ces deux inconvénients.

3.3 Notre modèle de C^2-processus

Le commandement et contrôle est en fait une tâche hautement cognitive qui necessite d'être développée à partir de l'analyse du comportement des décideurs lorsqu'ils sont en face de différentes situations: routinières, familières, et non-familières (ou de crise). C'est pourquoi, nous pensons que l'architecture proposée dans le cadre de cet article peut contribuer à modéliser convenablement le commandement et contrôle considéré comme un process. Ce nouveau processus est represénté en figure 4 où l'information, à la différence du modèle de Lawson peut suivre l'un des chemins suivants: (1) l'information est perçue en terme d'action; (2) l'information est perçue en terme de but ou de ressource à planifier; (3) l'information est identifiée ou reconnue en terme d'action; (4) l'information est identifiée ou reconnue en terme de but ou de ressource à planifier; (5) l'information reflète un engagement dans une action. Finalement, la base de connaissances et de croyances est reliée aux cinq autres modules par (6).

En fait, l'architecture proposée a l'avantage d'être issue de l'analyse des activités cognitives effectuées habituellement par les décideurs en cas de crise, et par conséquent elle reflète mieux la réalité. D'ailleurs, cette architecture répond mieux aux situations contraintes par le temps et plus particulièrement aux situations réflexes qui nécessitent des temps d'action très rapides. C'est le cas par exemple, d'un pilote d'avion de chasse qui voit au dernier moment venir sur lui un missile. On voit alors mal. comment il pourrait suivre les différentes étapes du modèle de Lawson. Dans notre modèle au contraire, ce pilote a un comportement reflété simplement par *perception ⟶ action*.

Dans le même ordre d'idée, le modèle que nous proposons reflète les différentes tâches cognitives réellement effectuées par les agents dans le cadre du commandement et contrôle. Ainsi par exemple, dans le cadre d'un poste de commandement NORAD (du type hiérarchique), des agents du type Intelligence, Weather et Fighter ont pour principale tâches: *per-*

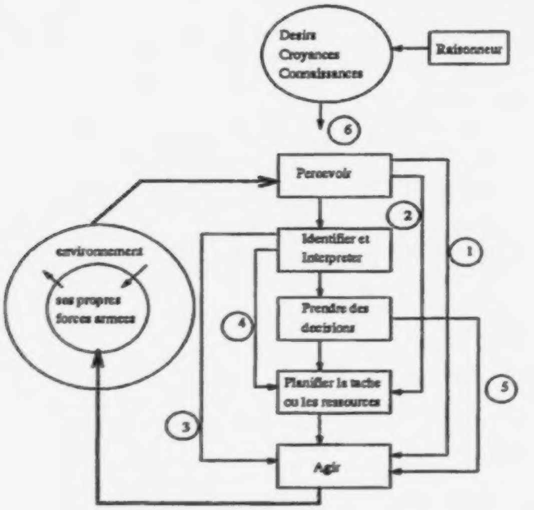


Figure 4: Notre modèle de C^2-processus

ception et *action*. Ces agents communiquent ensuite les informations perçues aux agents Director of Ops et Senior Director qui utilisent dans la plupart des cas les étapes: *perception*, *reconnaissance et identification*, *prise de décision* et *action*. Une fois leurs décisions élaborées, ils les communiquent au Commandement qui effectue généralement les étapes: *perception*, *reconnaissance et identification*, *prise de décision*, *planification* et *action*.

Il convient de préciser aussi, que d'une façon générale le C^2-système est une organisation d'agents interagissant soit dans le cadre d'une organisation militaire hiérarchique soit dans le cadre d'une *coopération* ou d'une *négociation* entre gradés d'un même niveau en vue d'allouer des ressources, ou en vue de trouver un compromis quelconque. Notre architecture permet de modéliser ce genre d'interactions puisqu'elle utilise le concept de carte cognitive et la planification multiagent. La carte cognitive permet de prendre des décisions dans tout environnement distribué, et le module de planification est capable d'élaborer des plans multiagents. Dans le cadre des interactions entre militaires, il exite plus particulièrement deux situations typiques, aisément modélisables par notre architecture. La première est celle d'un commandant de région ayant à sa disposition des informations relative à une vaste région, et un régiment de subordonnés à qui il envoit des ordres. Pour représenter cette hiérarchie, on pourra utiliser le modèle de la figure 4 de manière récursive de façon à avoir des C^2-processus montés en "nids". Une telle organisation hiérarchique est représentée en figure 5.

La seconde situation typique aux environnements militaires est celle où un commandant de région a sous ses ordres des subordonnés, dont les environnements sur lesquels ils agissent, se chevauchent. L'objectif du commandant consiste alors à *coordonner* les subordonnés afin de tirer le maximum de leurs activités tout en évitant les redondances et les incidents ou accidents que peuvent induire les chevauchements. Ce type de coordination peut aussi se faire en utilisant le modèle de la figure 4, puisque ce modèle permet à chaque agent de construire sa carte cognitive en composant plusieurs autres cartes individuelles de manière à avoir une vue globale. Ceci peut très bien s'appliquer au commandant de région qui doit dans ce cas, avoir une vue générale englobant les parties qui se chevauchent. A ce propos, une vue globale de ce type de coordination est donnée en figure 6.

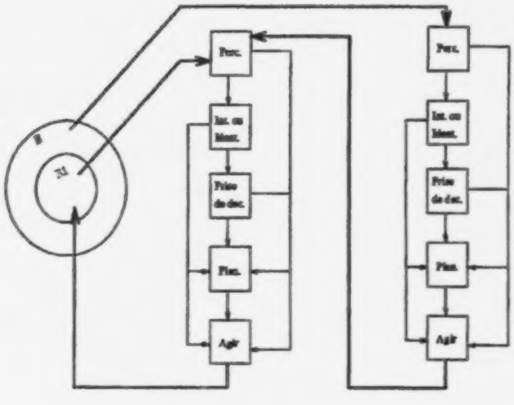


Figure 5: C^2-procces en "nids"

Bien entendu, les éventuelles possibilités d'utilisation du modèle C^2-processus proposé sont nombreuses. C'est ainsi que le modèle proposé peut servir entre autres à:

- simuler des plans de bataille, des stratégies de défense, des politiques d'allocation des ressources, etc.
- former les officiers, sous-officiers ou autres à prendre des décisions face à toutes les situations (normales, habituelles et de crise);
- concevoir des systèmes d'aide à la décision;
- concevoir des systèmes intelligents complètement autonomes (robots, télérobots, avions intelligents, engins militaires intelligents, etc.) capables de remplacer les hommes (completement ou partiellement) dans les environnements hostiles (e.g. détection et destruction de mines par exemple).

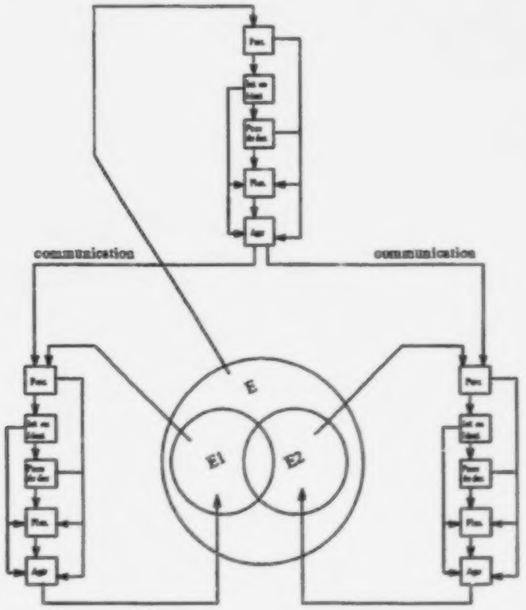


Figure 6: Coordination entre différents C^2-processus

4 Implantation sur machine

L'architecture proposée, muni de ses différents mécanismes de raisonnement, a été implantée en partie sur machine SunSparc en utilisant le langage Common Lisp. Les composantes de planification et d'identification n'ont pu être implantées *faute de temps*. Toutefois, ces deux composantes ont été "simulées" (respectivement par un ensemble de plans et de cas) de façon à compléter l'architecture et voir ses différents comportements face à différentes situations et en présence de plusieurs agents.

Cette implantation a donner lieu ensuite à une simulation d'environnement multiagent comprenant plusieurs véhicules intelligents capables d'interagir entre eux. Cette interaction est essentiellement une coexistence dans laquelle les agents arrivent à éviter tout incident facheux (collision, blocage, etc.) grâce aux connaissances "sociales" intégrées au niveau de chacun des agents. Ce type d'exemple est particulièrement intéressant et il peut facilement être adapté à plusieurs autres domaines du commandement et contrôle:

- interaction entre plusieurs avions "intelligents",
- interaction entre blindés et autres véhicules "intelligents",
- interaction entre robots et télérobots "intelligents",
- interaction entre plusieurs décideurs qu'ils soient des machines "intelligentes" ou des êtres humains.

5 Conclusion

Rappelons qu'il existe en fait deux approches pour aborder les environnements multiagents qui sont des environnements fort complexes et dans lesquels la coopération et plus généralement l'interaction entre plusieurs processus C^2 s'intègre très bien. La première est une approche "du haut vers le bas" et elle consiste à s'intéresser plus spécialement au groupe d'agents. C'est ainsi que sont étudiés et approfondis des concepts tels que: coopération, cohésion, coordination, négociation, communication, etc. La seconde approche est une approche "du bas vers le haut" et elle consiste à étudier l'architecture et le comportement d'un agent individuel évoluant dans un environnement multiagent. Le but alors est de donner à chaque agent les capacités sociales de façon à lui faciliter l'interaction avec les autres agents.

Notre travail s'est inscrit dans cette seconde approche afin de modéliser tout d'abord C^2 vu comme un processus, et de voir ensuite comment on pourrait aborder l'interaction entre plusieurs processus de C^2. Dans ce contexte, nous avons présenté une architecture d'agent basée sur les comportements tels qu'on les trouve dans les environnements réels multiagents et plus particulèrement dans les organisations hiérarchiques militaires. De plus, cette architecture a été conçue en vue de tenir compte des intentions des autres agents tant au niveau de la prise de décision qu'au niveau de la planification basée sur les cas. Nous lui avons aussi intégré des connaissances sociales telles que les normes sociales, les règles coopératives, etc. de façon à enrichir les niveaux "routines" et "situations familières", et permettre ainsi à chacun de coordonner ses activités avec les autres agents. Un exemple de coordination où cette architecture pourrait s'appliquer est celui d'un commandant de région qui tente justement de coordonner les activités de ses subordonnés, comme cela a été détaillé en 6.3.

Bien entendu, l'architecture proposée est loin d'être terminée et il convient de l'étendre de façon à au moins prendre en compte:

- la génération des actes du langage naturel de façon à communiquer à autrui ses intentions et ses désirs,
- la compréhension des actes du langage naturel de façon à raisonner sur autrui pour coordonner ses activités et assurer une certaine cohérence.

Remerciements

Ce travail a été rendu possible grâce au support financier du Conseil de Recherches en Sciences Naturelles et en Génie du Canada (CRSNG), du fonds pour la Formation de Chercheurs et l'Aide à la Recherche de la province de Québec et du Centre de Recherches pour la Défense de Valcartier (CRDV).

References

[1] R. Axelrod (ed.). ***Structure of Decision: The cognitive Maps of Political Elites.*** Princeton University Press, 1976.

[2] B. Chaib-draa, E. Paquet and L. Lamontagne. Integrating Reaction, Planning and Deliberation in Architecture for Multiagent Environment. *Proc. 3rd Conf. on Computer Generated Forces and Behavior Representation, Orlando, Florida*, 1993.

[3] B. Chaib-draa, E. Paquet and L. Lamontagne. Architecture d'un Système Intelligent pour les Environnements Multiagents. *Actes des Journees Volcan-IAA 93, Clermont-Ferrand, France*,1993.

[4] J. Lawson, Jr., Command-Control as a Process. *IEEE Contr. Syst. Magazine*, vol.16, 1981.

[5] M.L. Metersky. A C^2 Process and an Approach to Design and Evaluation. *IEEE Trans. Sys. Man and Cyber.* SMC-16(6), 1986, pp.880-889.

[6] J. Rasmussen. *Information Processing and Human-Machine interaction: An Approach to Cognitive Engineering*, North Holland, 1986.

[7] R.C. Schank. ***Dynamic Memory: A Theory of Reminding and Learning in Computers and People.*** Cambridge University Press, 1982.

905C20

Floor Obstacle Avoidance for a Mobile Robot[*]

A. Manz, D. A. Green, S. Elgazzar, F. Blais and M. Lecavalier [†][‡][§]
Autonomous Systems Laboratory
Institute for Information Technology National Research Council of Canada
Montreal Rd, Ottawa, Ont. K1A 0R6

#149740

Abstract

Reliable operation of an autonomous mobile robot is highly dependent upon a robust navigation system that can cope with the complex nature of uncertain environments. Because the problem is very complex, the implementation of such a system is facilitated by identifying distinctive segments of the overall problem which can initially be solved in isolation but later incorporated into a more comprehensive solution.

One segment of the autonomous navigation problem which has been identified is real time collision avoidance for an indoor enviroment. In this paper the initial implementation of a system capable of safely navigating a mobile robot in the presence of low profile floor obstacles and floor discontinuities will be examined. This system builds upon our earlier work on reactive collision avoidance that uses sonar data and a grid representation to detect and avoid obstacles. The system being developed at the Institute for Information Technology of the National Research Council of Canada is based on the use of a multi-headed BIRIS[¶] laser range measuring system.

The results of a simulation of the collision avoidance system are presented. This simulation was used to determine a configuration in which three 30° field of view BIRIS sensor heads would adequately sense the floor surface. The method used to provide the coordinate transformation between the sensor coordinate system and the robot coordinate system is also described. An evaluation of the performance of the BIRIS system using one sensor head configuration is presented. Finally, some preliminary results of the operation of a mobile robot equipped with a single BIRIS floor sensor is described.

Introduction

The desire to limit exposure of personnel to hazardous situations, to improve the effectiveness of skilled employees and to reduce operating costs has created a demand for autonomous robots. One task an autonomous mobile robot is required to accomplish is navigation through a complex and possibly evolving indoor environment about which no prior information may be available. Reliable navigation of a mobile robot through such an environment is required because an autonomous robot is expected to navigate with little or no human supervision.

Implementation of a complete autonomous navigation system presents a very difficult problem. The implementation process can be facilitated by identifying distinctive segments of the overall problem which can initially be solved in isolation but later incorporated into a more comprehensive solution. One segment of the autonomous navigation problem which has been identified and which we are pursuing is real-time collision avoidance. Real-time refers to the ability of the mobile robot to avoid obstacles without hesitation using on-board sensors. The problem of real-time collision avoidance can in itself be broken down into different parts. The decomposition of the real-time collision avoidance problem is based on the type of obstacles to be avoided, the desired response of the navigation system to detected objects and the rate at which the complete system can respond to recently detected objects. These factors influence the type and number of sensors used, the manner in which raw data is used to model the environment and the kind of algorithms employed to interpret the environmental model and to navigate the robot.

The design objective in the existing implementation of a real-time collision avoidance system [5] at the Institute for Information Technology of the National Research Council (IIT) was to avoid unexpected objects, such as walls, posts, chairs, tables, etc., using only sensory information. This system was to perform this function while attempting to achieve a limited goal, which was to travel either in a desired direction or to a global location, with minimal deviation in the original speed and path of the robot. We call this "reactive" navigation. This differs from "reflexive" navigation where the desired goal is abandoned in the presence of an unexpected obstacle. We distinguish both of these from "path planning" navigation which is concerned with guiding the robot through a static environment but which is not concerned with collision avoidance because of its slow reaction time.

---

[*] NRC No. 37096

[†] Tel: (613) 993 - 7889

[‡] Fax: (613) 952 - 0215

[§] e-mail: *last name*@iit.nrc.ca

[¶] Official Mark of the National Research Council of Canada

In this paper the initial implementation of a system capable of safely navigating a mobile robot in the presence of low profile floor obstacles will be examined. This system complements and extends upon our earlier work on reactive collision avoidance that used sonar range data and a grid representation to detect and avoid obstacles. A mobile robot equipped with this capability will be able to detect and avoid obstacles on the floor that are either elevated or depressed. In the future, it is planned to construct a more sophisticated system that will be able to classify objects into those that can be safely ignored, those that can be conditionally traversed in a very specific manner, and those that must be avoided.

Following this introduction, an overview of the existing implementation will be presented. This will be followed by a discussion of the issues regarding the selection of a suitable sensor for detecting floor obstacles. Included with this discussion is a description of a sensing system that has been selected. The next portion of the paper describes a computer simulation used to determine an acceptable configuration for the sensor heads of the sensing system. The discussion includes a description on how the sensing of the environment and the performance of the mobile robot were modelled. It also contains a brief description of the mapping and navigation algorithms used in both the simulation and physical implementation of a collision avoidance system for floor obstacles. The insights gained through the use of the simulation will also be presented. In the next section of the paper a physical implementation of a collision avoidance system will be described along with the results from some preliminary tests. This description includes the method used to transform the sensory data into platform coordinates. The last section of the paper will contain a brief summary of future work.

Overview of Existing Implementation

The initial implementation of a collision avoidance system was carried out using a Cybermotion K2A mobile robot [9], a ring of 24 sonars for range measuring [4] and a multiprocessor computer system using the Harmony[‖] operating system [3]. The K2A robot is a commercially available robot that utilizes a unique concentric shaft drive system which provides synchronous steering and propulsive movement to each of the three wheel assemblies. This drive system allows a single steering motor and a single propulsion motor to control all wheel assemblies in such a way that all wheels point in the same direction at all times. As a result of this design, the base of the robot does not rotate as it executes turning manoeuvres and allows the robot to perform a zero radius turn. However, the turret on which the BIRIS sensor head is mounted is directly connected to the wheels through the concentric shaft system and thus always points in the same direction as the wheels.

Reactive navigation was achieved by a system using the ring of sonars to sense an annular volume about the waist of a mobile robot. The sonar range measurements were used to construct a two dimensional Cartesian grid representation of the environment about the robot. The certainty grid method [2, 5, 6] uses temporal integration of successive range measurements to minimize the effects of the spurious data typically produced by sonar range sensors. The sector elimination method was then used to select a direction for collision-free motion which most closely matched the desired direction of motion. The sector elimination algorithm divided the sensed environment into 48 sectors centred about the robot. Obstacle information that was extracted from the Cartesian map was used to determine which of these sectors was occupied by an object. These occupied sectors and a number, which was dependent on the distance between the object and the robot, of adjacent sectors were eliminated from the courses in which the robot would be steered. The eliminated sectors were labelled occupied sectors and the remaining sectors were labelled open sectors. The robot was steered in the direction of the bisector of an open sector which matched the desired direction of travel as closely as possible. This system, when placed in a "wandering" mode, is able to safely roam through an area containing randomly scattered stationary or slow moving obstacles. Using the sonar ring, this method has been successfully demonstrated. However, it is incapable of detecting and thus avoiding low profile objects on the floor or discontinuities in the floor.

Sensing

Sensing and sensory data interpretation are critical elements in creating a collision avoidance system. Ideally, a volume enveloping the entire mobile robot should be continually sensed. The rate at which this data can be gathered and interpreted must be such that the appropriate control instructions can be delivered to a robot in a timely fashion. Unfortunately, no single sensor is capable of both gathering all the desired information at the required rates or presenting it in a format that can be interpreted quickly enough for real-time collision avoidance. Therefore, to create a robust collision avoidance system multiple sensors and multiple types of sensors will be required to complement the capabilities of each other.

Sensing Requirements

One way to improve the robustness of the existing collision avoidance system is to give it the ability to sense and avoid obstacles on the floor. Two requirements for the sensing system that had to be addressed were the distance that must be sensed in front of the robot and the continuous need to

---

[‖] Harmony is a mark reserved for the exclusive use of Her Majesty the Queen in right of Canada by the National Research Council Canada.

obtain data representing the floor. The later requirement is necessary in order to detect obstacles such as stairs, channels and other depressions in the floor. An initial estimate of the required resolution of the sensing device is based on the size of the tires of the mobile robot. The width and diameter of the tires, 3.5 cm and 20.5 cm respectively, makes it reasonable to assume that the robot could cross any type of obstacle which is less than 2.5 cm in its largest dimension at any orientation. The rate at which data must be gathered and processed is dictated by two factors. One factor is the maximum speed at which the mobile robot is expected to operate. The second factor is the extent of the environment that can be sensed during one sample.

Sensor Selection

The nature of floor sensing severely restricts the type of sensing device which would be appropriate. Sonars were not considered because of their low angular resolution and susceptibility to specular reflections. Stereo vision has also been eliminated as a possible type of sensing device because of its large computational requirements. Finally, the short sensing ranges, 1 m to 3 m, makes time-of-flight devices using light inappropriate. The capabilities of structured light devices to measure distances came closest to fulfilling the requirements for floor sensing. The structured light range measuring system that was selected for the collision avoidance system is called BIRIS [1]. BIRIS was developed at the IIT.

The BIRIS range measuring system is composed of one or more sensor heads and dedicated data processing. The BIRIS head uses a standard CCD video camera, a laser line projector, and a modified lens. A mask with two apertures replaces the iris of the camera lens (Bi-IRIS), see Figure 1, which produces a double image of the projected laser line. The separation of the double-imaged lines measured, A - B, on the CCD array of the camera is inversely related to the distance from the projector to the object. The position of the centroid of the double image, (A+B)/2, is directly related to the range when the projected plane of laser light is offset from the axis of the camera. A real-time dedicated processor computes the range and intensity data at video rates. This data is communicated as the range from the sensor, z value, and the lateral offset from principle axis of the sensor, x value.

This arrangement offers a number of advantages: high immunity to ambient illumination and background interference due to its ability to discriminate false measurements, compactness and reliability, good resolution and speed, and the potential for low cost implementation.

Characteristics of the current BIRIS range measuring system are as follows:

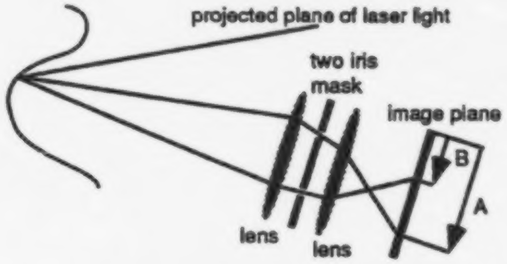


Figure 1: *BIRIS Schematic*

Measurement range:	0.4 - 4 m	
Accuracy (rms):	1.6 mm @ 0.5 m 2.5 mm @ 1 m 4.2 mm @ 2 m	
Focal length:	17 mm	
Field of view:	22/30 degrees	
Acquisition time:	50 ms	33 ms
Points per line:	128	64
Laser power:	16 mW	
Eye safe distance:	>16 cm	

Sensor Data Utilization

In order to use a sensor like BIRIS a perimeter must be established around the mobile robot through which all stationary obstacles must pass as the mobile robot moves. The smallest detectable obstacles that pass through this perimeter will be detected only once. Also, each portion of a larger obstacle will be scanned only once. Therefore, it becomes imperative that the data generated by the sensor be relatively error free. As a result the probabilistic idea used to update a Cartesian grid in the certainty grid method, which had been successfully used in the existing implementation of the collision avoidance system to minimize the effects of sensing errors can no longer be applied. The Cartesian grid map representation which we are using contains only two values, occupied and unoccupied, as opposed to the certainty grid which contains a range of values indicating the probability of occupancy. However, to further investigate what portion of the enviroment has been sensed by the BIRIS system, unoccupied cells representing areas which have been sensed and those that have not been sensed will be differentiated.

Another implication of using a structured light sensor, which uses a single plane of light, is that a representation of the environment must be used to compensate for the extremely limited portion of the environment that is sensed during each acquisition. This representation will have to integrate information from successive acquisitions from multiple sensor heads in order to construct a more complete understanding of the environment and it will have to adjust the position of the obstacles relative to the mobile robot as the robot moves.

Determining the number, location and orientation of BIRIS heads required to establish the desired sensing perimeter around the robot is difficult. In order to facilitate this process and to experiment with different variations in the two dimensional mapping and navigation algorithms, a simulation of the collision avoidance system was developed.

Simulation

A simulation of the collision avoidance system for floor obstacles was used to ascertain the effectiveness of the whole system. Therefore, it was important to simulate as much of the system as was practical because of the interdependencies between the various components of the whole system. The three major elements that are to be assessed by the simulation are the portion of the environment that would be sensed by the BIRIS system, the mapping and navigation algorithms that would be used to generate the control signals and the response of the robot to these signals.

The simulation is used to determine the minimum number of BIRIS sensor heads required to sense all the areas over which the robot would traverse. Additionally, the simulation is used to further refine the sensor head configuration by illustrating through a graphical interface the effects of changing the mounting position, the orientation or the field of view of each BIRIS sensor head.

The graphical interface displays the intersection of the floor and the plane of laser light at each sensing interval, the track of the robot tires, a ring indicating the maximum physical extent of the mobile robot and user specified point obstacles. The simulation is able to create a three dimensional model of the robot and the planes of laser light emitted by each sensor head so that it can be ascertained if portions of the robot would obstruct the view of any BIRIS sensor head.

Sensor Pose Considerations

Figure 2, shows examples of the simulation output and illustrates the difference in the density of the scanned area for two different camera configurations for a constant rate turn while maintaining a constant linear speed. Robot dynamics were not modelled. In Figures 2a and 2b the scan patterns are the same but the distance from the centre of the robot to the periphery of the scanned area is increased from 0.9 m to 1.2 m, respectively. Figure 2 illustrates some of the conflicting requirements for the sensing. The larger the distance from the robot to the periphery of the scanned area the greater the time available for the robot to respond to newly encountered obstacles and the greater the area sensed during each data acquisition. Unfortunately, the density of the scans within the scanned area and the resolution of the data decreases as the distance to the periphery of the scanned area from the front of the robot increases. The increase in the scanned area during each acquisition and the decrease in the resolution is a result of the sensing system measuring range data at evenly spaced angular positions within the field of view. To highlight the differences in the sampling density for the two configurations only every fourth simulated scan line appears in Figure 2.

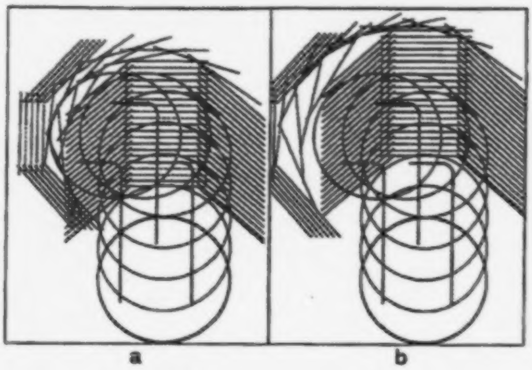


Figure 2: *Configuration Comparison between: a) 0.9 m range b:) 1.2 m range*

Sensor Data Simulation

The simulated sensing interval between acquisitions of BIRIS data was specified as 50 ms which is the same for the physical BIRIS system. Therefore, for the three camera configurations being simulated, each camera provides information every 150 ms. Each BIRIS data acquisition consists of an array of 128 range samples and the associated x position in the coordinates of each camera. At each acquisition the intersection of the laser plane and the floor was calculated. The coordinates for 128 evenly spaced points along the intersection line and the coordinates of any obstacles that were within 1.25 cm of this line are sent to the navigation algorithms being tested.

Navigation Algorithms

The mapping algorithm uses the coordinates of each detected obstacle and each of the evenly spaced point on the intersection line to assign a value to the cell corresponding to that location within the robot-centred Cartesian grid map. All unscanned grid cells are initialized to one. As the robot moves, the grid cells representing areas that have been scanned are assigned a value of negative one if no obstacles are detected or zero in an obstacle is detected. This Cartesian map was displayed on top of the previously described graphical interface with the value of each cell being indicated by a different colour. The square Cartesian grid map represents an area 2.2 m in width with the individual cells within the map representing areas 2.5 cm in width.

To find a direction of collision-free motion for the mobile robot a polar grid map, consisting of 72 sectors, is overlayed onto the Cartesian grid map. At the sampling rate of the mobile robot, 100 ms, the range of the closest obstacle within each sector was extracted from the Cartesian map. For collision free navigation, some number of sectors adjacent to sectors containing obstacles must also be labelled as occupied. This number was arrived at heuristically and increases as the range to the objects decreases. The remaining sectors were labelled open. The mobile robot was steered in the direction of the bisector of the open sector which matched the desired direction of travel as closely as possible. The steering rate input, V_α was calculated using the angular difference between the desired direction of travel and the current direction of travel, P_α. The steering rate was computed using:

$$V_\alpha = \begin{cases} 2 \times \Delta P_\alpha & \text{if } V_\alpha < V_{\alpha max}, \\ V_{\alpha max} & \text{otherwise,} \end{cases} \tag{1}$$

where:

$V_{\alpha max}$ is the maximum allowed steering rate.

Using a gain factor larger than two resulted in highly oscillatory steering motion. The heuristically arrived at equation to calculate the linear speed input, V_d, to be used by the simulation was:

$$V_d = V_{dmax} \exp^{-2V_\alpha}, \tag{2}$$

where:

V_{dmax} is the maximum allowed linear drive rate.

This equation allowed the robot to maintain a high speed for small steering rates but drastically curtailed the linear speed as the steering rates increased.

Robot Simulation

The model of the robot response was developed from experimental data. The synchro-drive mechanism of the K2A Cybermotion mobile robot is such that the drive and steering system can be assumed to be independent. Sinusoidal input signals, in the form of desired velocities, of different periods but constant magnitude were used as control signal inputs to the driving and steering sub-systems of the robot. The odometric output of the robot and input signal were then used to create Bode plots for each robot sub-system. Based on the information available from the Bode plots a linear transfer function for each sub-system was estimated, plotted, and compared with the experimental data. These functions were then modified until a close approximation to the experimental data was reached. For the steering system the estimate of the linear transform is:

$$G(s) = \frac{e^{-0.32s}}{s(0.09s+1)}. \tag{3}$$

For the drive system functions the estimate of the linear transform is:

$$G(s) = \frac{8^2 7^2 e^{-0.07s}}{s(8^2 + 2(0.6)8s + s^2)(7^2 + 2(0.5)7s + s^2)}. \tag{4}$$

The appropriate difference equations were calculated using these linear transfer functions for a sampling period of 0.01 s and used to model the response of the mobile robot to the simulated input signals.

Simulation Results

An example of the simulation results is presented in Figure 3. The scale of the world grid is 20 cm and the dots represent point obstacles. The lines in this figure represent the portion of the environment that was sampled by the BIRIS system. For clarity, a line was drawn only for every seventh simulated acquisition. The individual values contained in the robot-centred Cartesian grid map are presented in the upper right hand portion of Figure 3. Black cells represent occupied locations. White cells represent sensed areas containing no obstacles and grey cells represent unsensed areas. The Cartesian grid map represents only a portion of the total environment shown in Figure 3. The values contained in the Cartesian grid map represented the view the navigation algorithm had of the environment during the last time step of the simulation.

The most significant observations obtained from the simulation concerns the treatment of unsensed areas and the minimum size of obstacles that can robustly be sensed. Initially, the navigation algorithms were to treat unsensed areas the same as sensed obstacles. Therefore, the resolution of the Cartesian grid was set to the size of the smallest obstacle to be detected, which was 2.5 cm. When the simulation was run with a two camera configuration, a maximum linear speed of 15 cm/s and a maximum steering rate of 70 °/s the number of isolated unsensed areas was significantly greater than those displayed in Figure 3. A two camera configuration was used because each sensor was then sampled at a faster rate, every 100 ms rather than every 150 ms for the case of a three camera configuration. These isolated areas were not scanned because of a combination of the sampling rate of the sensing system and the increased relative velocity of the floor with respect to the line along which each sensor head could sense as the robot was steered. Additionally, if the entire swath over which the robot was to pass was to be scanned a portion of the environment could never be sensed while using a two 30° field of view sensor heads because of the geometry of the situation. This applied to all turns in which the robot was moving slowly or stopped. Turning while maintaining the same position was required in most circumstances because of the extremely limited view of the environment. This lead to the use of a three sensor head configuration in which the field of view for each head was 30°. The sensing rate and desired robot speed also lead to the temporary abandonment of the 2.5 cm maximum undetected obstacle size.

In the simulation all the unsensed areas were treated as unoccupied areas. During subsequent simulation trials it became evident that unless the maximum linear speed and maximum steering rate were reduced to unacceptably low magnitudes the appearance of isolated unsensed areas (as opposed to areas which never come within the field of view of any sensor head) in the Cartesian map was going to occur. As a result, it was decided to treat isolated unsensed area the same as sensed but unoccupied areas. Also, the positioning and orientation of the

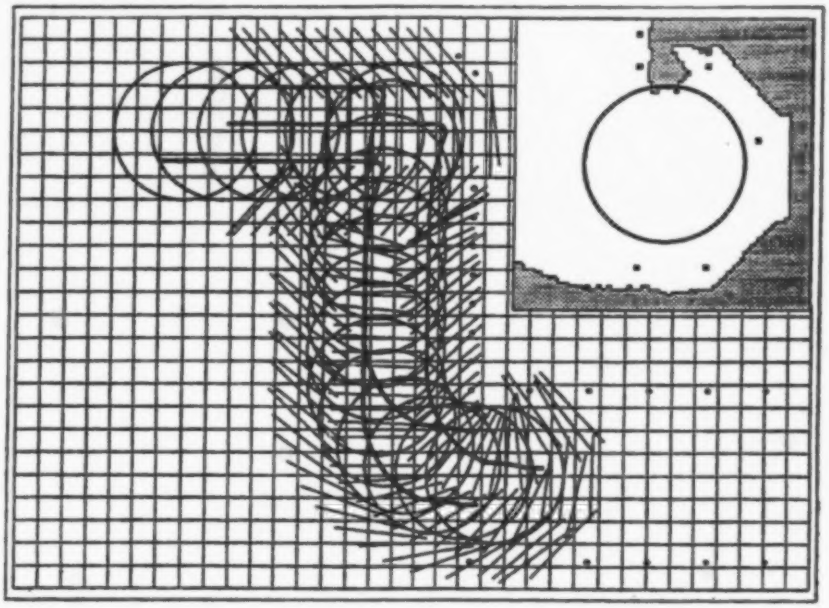


Figure 3: *Simulation Results*

sensor heads guaranteed that the area over which the robot was to traverse came within the field of view of at least one sensor head. Therefore, treating all unsensed areas within the Cartesian gird map as unoccupied seemed justified.

Implementation

The initial experimental version of the collision avoidance system was constructed using a single sensor head BIRIS system because the design of the software and the rate at which data were provided were not dependent on the number of sensor heads. Although, increasing the number of sensor heads decreases the rate at which any one head is used to sample the environment. Therefore, it was possible to analyze the performance of the range measuring system in situ which allows final refinements to the configuration design of the sensor head to be made before all three sensor heads were constructed.

Coordinate Transformation

In order to use a range measuring system, a method for transforming the two dimensional coordinates, x and z, of the camera into three dimensional robot coordinates, x, y, and z, had to be determined because objects detected in camera space had to be mapped into a robot centred Cartesian map. The method required that three non-collinear points be measured in camera coordinates and turret coordinates, P_0, P_1 and P_2 in Figure 4. The origin of turret coordinate system, O_t, was set to coincide with the origin of the robot coordinate system. However, the orientation of the turret coordinate system changes as robot steers while the orientation of the robot coordinate system remains fixed with respect to a global coordinate system.

The turret coordinate points were obtained by physically measuring, with respect to the robot and the direction of the turret, the locations at which the laser light from a BIRIS head impinged on edges of an object. This object consisted of two prisms joined along one edge. The resulting shape measured by the BIRIS system resembled an inverted W. The four lines were extracted from the range data using a random sampling method [7]. The intersection of these four lines corresponds to the locations which were physically measured. The effects of random range measurement noise can be reduced and the inaccuracies due to the angular resolution of the BIRIS sensing system can be minimized by extracting lines and calculating intersection points rather than trying to use a single point from the range data.

The transformation method used a 4 × 4 coordinate transformation matrix [8]. This matrix was determined by first calculating the unit vectors for the axis of the camera coordinate system in the turret coordinate system. This 3 × 3 matrix, R_t^c, was augmented to a 4 × 4 matrix, T_t^c, to incorporate the displacements of the origin of the turret coordinate system from the origin of the camera coordinate system. The T_t^c coordinate transformation matrix was multiplied by a turret to robot coordinate transformation matrix, T_r^t to obtain a camera to robot transformation matrix, T_r^c.

The elements of the R_t^c were determined by first calculating the direction of the camera y axis in turret coordinates. In the camera coordinate sys-

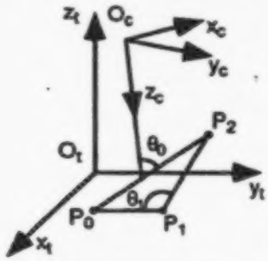


Figure 4: *Camera and Turret Coordinate Systems*

tem the y axis was perpendicular to the plane of the laser light. Therefore, the cross product of two lines in turret coordinates, $\overline{P_0P_1}$ and $\overline{P_2P_1}$ in Figure 4, created using the three points in turret space, defined the direction of the camera y axis in turret space. It was necessary to normalize this vector because its magnitude was dependent on the size of θ_1. The next step in the process was to calculate the angle, θ_0, using a dot product between the z axis of the camera coordinate system and a line, $\overline{P_2P_0}$, using camera measurements. Three simultaneous equations can be obtained for the various components of the z axis of the BIRIS sensor head in turret coordinates using:

$$\vec{y_{ct}} \sin\theta_0 = \overline{P_{2t}P_{0t}} \times \vec{z_{ct}}, \tag{5}$$

where:

$\vec{y_{ct}}, \vec{z_{ct}}$ are unit vectors in the direction of the y and z axes of the camera in turret coordinates,

$\overline{P_{2t}P_{0t}}$ is the line formed between P2 and P0 in turret coordinates, and

θ_0 is the angle between the z axis of the camera coordinate system and $\overline{P_{2t}P_{0t}}$.

Obtaining these three equations was possible because the magnitude of θ_0 will be the same in both coordinate systems. Unfortunately, these equations will always be linearly dependent. Therefore, one of these three equations had to be replaced. The dot product of $\overline{P_{2t}P_{0t}}$ and $\vec{z_{ct}}$, which equals $\cos\theta_0$, provided the desired linearly independent equation. The equation that was replaced was arbitrary except in situation in which the line $\overline{P_{2t}P_{0t}}$ was parallel to one of the axes of the turret coordinate system. In these cases, the equation that was replaced was the one equation out of the three obtained using the cross product that had only one non-zero component. The unit vector along the x axis of the camera in turret coordinates was calculated using the cross product of the previously calculated y and z axis unit vectors.

To create the T^t_c coordinate transformation matrix, T^t_c, the location of the turret coordinate system origin, D^c_{xt}, D^c_{yt} and D^c_{zt}, with respect to the camera coordinate system origin, O_c, was calculated. These displacement values were calculated using a measured position of a point in camera coordinates and the measured position of the same point in turret coordinates and the unit vectors of the camera coordinates axes defined in the turret coordinate system. A turret to robot coordinate transformation matrix, T^r_t, was multiplied by T^t_c to obtain the desired camera to robot coordinate transformation matrix, T^r_c. T^r_t was derived [8] for the case in which the origins of the robot and turret coordinate systems were always the same and their respective z axis always coincided. The only motion between the two coordinate systems was a rotation about the z axis, which was the steering orientation of the robot. T^r_c was equal to:

$$T^r_c = \begin{bmatrix} x_i c\phi - x_j s\phi & y_i c\phi - y_j s\phi & z_i c\phi - z_j s\phi & D^c_{xt} c\phi - D^c_{yt} s\phi \\ x_i s\phi + x_j c\phi & y_i s\phi + y_j c\phi & z_i s\phi + z_j c\phi & D^c_{xt} s\phi + D^c_{yt}\phi \\ x_k & y_k & z_k & D^c_{zt} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{6}$$

where:

x_i, y_i, z_i, x_j, y_j, z_j, x_k, y_k, z_k are the components of the x, y, z camera coordinate axes in the direction of the x, y, z turret coordinate system axes, denoted by i, j, k subsripts respectively, and

$c\phi$, $s\phi$ are the sine and cosine, respectively, of angle of the robot coordinate system with respect to the turret coordinate system.

The second column of T^r_c was never used because there was no y component to the data from the BIRIS system.

Range Data Utilization

The coordinate transform, T^r_c, was not applied to all the data produced by the BIRIS sensing system. The initial design of the collision avoidance system for floor obstacles assumed that the floor on which the robot operated was normally flat and level. Therefore, data describing a flat and level floor was redundant. The first step in utilising the range data was to extract from the range data only the data which did not conform to a flat and level operating surface. This was accomplished by using the random line extraction method [7] to find the data representing the floor and discarding them. The remaining data points were transformed using T^r_c and used to update the two dimensional robot-centred Cartesian grid map. The transformation was reserved for the last step because it was computationally cheaper to find a line in two dimension space as opposed to three dimensional space. Additionally, more computational time was saved by transforming only those data points which contained interesting information.

Once all the interesting data points had been identified, transformed and mapped into the two dimensional robot-centred Cartesian grid map, the navigation algorithms generated the appropriate robot control signals.

Implementation Performance

The implementation of a collision avoidance system for floor obstacles successfully guided the robot around sections of 2×4, 4×8×30 cm, placed directly on its path. The maximum speed at which the system was tested was 20 cm/s.

The sensing system provided range information over a varying angular range of approximately 20°. The discrepancy between the stated and actual performance is attributed directly to the sensed environment and the power of the laser projector. The lustre of a white linoleum floor severely reduced the amount of laser light reflected back to the sensor head which limited its performance. This had a significant impact on the treatment of the range data. The implementation treated gaps in range data along the line sensed the same as detected obstacles. An example of a large gap in range data indicating the presence of an obstacle is encountered when measuring holes in grates that cover larger openings. In order to implement this strategy the collision avoidance system determined the angular range over which range data was expected during an initialization routine. If this expected angular range of data was not received while the robot moved, the collision avoidance system would conclude that an obstacle was located close to one limit of the field of view of the sensor head. This caused unnecessary deviations by the robot away from the original path.

Future Plans

In the future a BIRIS sensor head will be constructed with a more powerful laser. Additionally, increasing the field of view for each sensor head to 35° or 40° is being considered to provide some additional flexibility in the positioning of the sensors.

Based on the performance of the complete sensing system, with the new sensor head configuration, a methodology will be developed to interpret gaps in range data correctly. Once the appropriate configuration for a sensor head has been ascertained, a collision avoidance system using three sensor heads will be implemented.

Long range goals for this project envision the recognition of conditionally traversable obstacles from the BIRIS range data and the execution of navigation strategies that would allow the robot to traverse such obstacles. Obstacles that fall into this conditionally traversable category are long narrow channels which are wider than the robot tires but significantly narrower than the diameter of these tires. This type of obstacle can be successfully traversed if approached in an appropriate manner.

Acknowledgments

This work was performed in cooperation with the PRECARN ARK (Autonomous Robot for a Known environment) Project, which receives its funding from PRECARN Associates. Inc., the Department of Industry Science and Technology Canada, the National Research Council of Canada, Technology Ontario and Ontario Hydro.

References

[1] Blais, F., Rioux, M., Domey, J., "Optical Range and Imange Acquistion for the Navigation of a Mobile Vehicle", IEEE International Confernence on Robotics and Automation, 1991, pages 2574-2580.

[2] Borenstein, J., and Koren, Y., "The Vector Field Histogram Fast Obstacle Avoidance for Mobile Robots", IEEE Transactions on Robotics and Automation, Vol. 7, No. 3, June 1991, pages 278-288.

[3] Gentleman, W. M., Archibald, C., Elgazzar, S., Green, D., and Liscano, R., "Case Studies of Real-Time Multiprocessors in Robotics", Proceedings of the Second International Specialist Seminar on the Design and Applications of Parallel Digital Processors", Lisbon, Portugal, April, 1991.

[4] Lang, S.T., Korba, L.W., and Wong, A.K.C., "Characterizing and Modelling a Sonar Ring", Proceedings of SPIE, Advances in Intelligent Robotics Systems:Mobile Robots IV, Philadelphia, Pennsylvania, November, 1989, pages 291-304,

[5] Manz, A., Liscano, R., Green, D., "A Comparison of Real- Time Obstacle Avoidance Methods for Mobile Robots", Proceedings of the Second International Symposium on Experimental Robotics, Toulouse, France, June, 1991.

[6] Matthies, L., Elfes, A., "Integration Of Sonar And Stereo Range Data Using a Grid-Based Representation", Proceedings of the 1988 IEEE International Conference on Robotics and Automation, Philadelphia, Pennsylvania, April, 1988, pages 727-733.

[7] Roth, G., Levine, M., Extracting Geometric Primitives, Computer Vision, Graphics, and Image Processing: Image Understanding Volume 58, July, 1993, pages. 1-22.

[8] Wolovich, W., Robotics: Basic Analysis and Design, CBS College Publishing, New York, New York, 1987, ISBN 0-03-006119-9, pages 45-99.

[9] K2A Mobile Platform, Cybermotion, Roanoke, Virginia, August, 1987.

2.93005

C-SPACE OBSTACLE COMPUTATION
A Geometry-based Fast Algorithm

C.S. Zhao*, M.M. Bayoumi* and M. Farooq**
* Dept. of Electrical Engineering, Queen's University, Kingston, Ont. K7L 3N6
** Dept. of Electrical Eng., Royal Military College, Kingston, Ont. K7K 5L0

#149742

Abstract

In this paper, we present a geometry-based fast algorithm for C-space obstacle computation. Obstacles could be either stationary or time-variant. Each obstacle in the robot workspace is modeled by a convex polygon. A discretized C-space representation is used for ease of implementation in order to build a subset of the Configuration space sequentially. C-space is considered as a matrix (C-space matrix). Instead of computing the C-space matrix element by element adopted by many researchers, we develop a fast algorithm to produce the C-space matrix block by block based on the use of some geometric properties. Due to the existence of time-varying obstacles, the C-space matrix is updated dynamically. Also a fast algorithm for C-space obstacle approximation is developed. For the purpose of the automatic collision-free motion planning, stationary obstacles and time-variant obstacles are treated differently. This is based on the fact that stationary obstacles always place constraints on the robot motion, while time-variant obstacles only impose constraints on the robot motion temporarily. A simulation is implemented for a planar, 2 degrees of freedom robot with revolute links.

1. Introduction

Developing autonomous robots represents one of the ultimate goals of robotics research. Such robots should be capable of accepting high-level descriptions of tasks and executing them without further intervention. A necessary step in developing autonomous machine controllers is the realization of on-line automatic collision avoidance, i.e. on-line collision-free path planning.

Most path planning schemes can be classified roughly into two categories: (1) explicit free space techniques and (2) potential function techniques. The first class of path planning algorithms builds explicit representation of subsets of robot configurations that are free of collision. Many algorithms based on this technique have been proposed [1][2][3]. In the potential function techniques, all algorithms are based on defining a penalty function on manipulator configurations which have the presence of obstacles [4].

For real-time robot control, path planning is commonly accomplished in the robot's configuration space. Implementation of automatic collision avoidance requires knowledge of obstacles within the machine's range of motion. However, computing forbidden regions within which motion of the robot is not allowed is quite a challenge in real time, especially for the robot with time-varying obstacles in its workspace. Most algorithms[1][3] use volume sweeping techniques to compute the forbidden regions. This can only be accomplished in an off-line mode, especially for robots with revolute links. Newman and Branicky [5] developed a real-time configuration space transform algorithm for obstacle avoidance. However, a lot of data base memory is required to store the tabulations precomputed in the off-line mode and the speed of transformation depends mainly on the number of primitives to be transformed.

In this paper, a geometry-based fast algorithm for C-space obstacle computation is presented. Efforts are made so that the C-space obstacle computation could be implemented in real time. We first describe the simulation environment and some assumptions. Next, in section 3 and section 4, the algorithm for computing and approximating the C-space obstacles is presented in detail. The final section deals with the conclusion and some discussions.

2. Background and Assumptions

We will restrict our work to the 2-D case. We assume that all obstacles in the work space can be modeled by convex polygons. Obstacles may be stationary or time-varying. For moving obstacles, they are assumed to be moving with a constant velocity.

A planar articulated robot with two revolute joints is chosen to illustrate the algorithm. Both joint axes are assumed to be parallel to the z-axis, so that the robot moves in xy plan. We choose this particular robot geometry based on the fact that it can illustrate many of the issues that arise with more complex industrial robots [9]. The positions of the robot joints are defined by the angles θ1 and θ2. Let l_1 and l_2 be the lengths of the first and the second link of the robot, respectively. The kinematics is given as

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} l_1\cos\theta_1 + l_2\cos(\theta_1+\theta_2) \\ l_1\sin\theta_1 + l_2\sin(\theta_1+\theta_2) \end{bmatrix} \tag{1}$$

$$\theta_1 = atan2\left[\frac{y(l_1+l_2\cos\theta_1) - xl_2\sin\theta_2}{x(l_1+l_2\cos\theta_2)+yl_2\sin\theta_2}\right] \tag{2}$$

$$\theta_2 = atan2\left[\frac{\pm\sqrt{1-\frac{(x^2+y^2-l_1^2-l_2^2)^2}{4l_1^2l_2^2}}}{\frac{(x^2+y^2-l_1^2-l_2^2)}{2l_1l_2}}\right] \tag{3}$$

where (x,y) is the position of the end-effector of the robot and (x,y) must satisfy the following condition

$$x^2 + y^2 \leq (l_1+l_2)^2 \tag{4}$$

In order to show the effectiveness of the proposed algorithm and to ensure that the method is more realistic, the parameters of the robot were chosen to mimic the IBM 7575 robot. i.e.

$$l_1 = 40.0\,cm; \qquad l_2 = 40.0\,cm$$
$$\theta_1 \in [-31°,\ 211°] \tag{5}$$
$$\theta_2 \in [-137°,\ 137°]$$

Due to the joint limits, the robot working space is not a torus. The radius of the inner circle is given by

$$r = \sqrt{l_1^2 + l_2^2 - 2l_1l_2\cos(\alpha)} \tag{6}$$

where $\alpha = 180° - \max(\theta_2)$. The simulation environment is shown in Fig.1. Obstacle #5 is a moving obstacle and the rest are stationary obstacles.

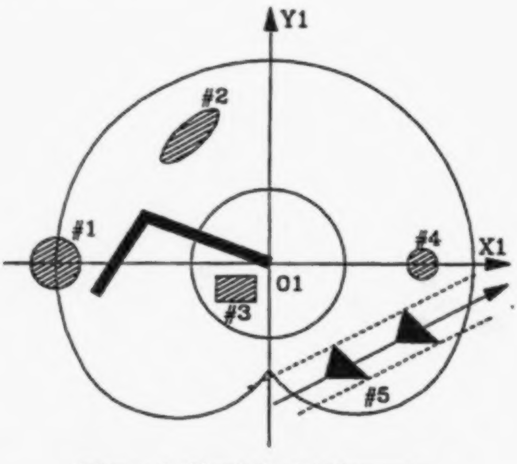


Figure 1 Simulation Environment

3. Forbidden Region Computation in C-space

3.1 Overview

Constraints placed on the robot's motion (joint limits and obstacles in the workspace) effectively reduce the volume of the valid configuration space. In the following, we will give the definition of the "*C-space obstacle*" caused by an obstacle.

Definition: The *C-space obstacle* caused by obstacle B in a robot's configuration space denoted as CO(B) is given by

$$CO(B) = \{x \in C \mid (R)_x \cap B \neq 0\} \tag{7}$$

This means that if $x \in CO(B)$, then $(R)_x$ intersects B. Here $(R)_x$ denotes the robot in configuration x, and $C \subset R^2$ is the configuration space for the given robot. Equivalently, the C-space obstacle caused by obstacle B is the set of all configurations which can cause a collision between B and some links of the robot.

It is noted that the *C-space obstacle* caused by a given obstacle can not be obtained simply by mapping the corresponding obstacle from the Cartesian space to the Configuration space. This is illustrated in Fig.2.

The regions A_1 and A_2 are not part of the obstacle A, yet these two regions must be considered as forbidden regions because they give rise to collision between the second link and the obstacle A. Regions in the workspace with such a property are called *shadows* of the corresponding obstacle [7]. Depending on the lengths of the links, the location, shape and size of an obstacle, there may be more than one shadow, or there may be no disconnected shadows at all. As a result, it is very difficult to characterize the shadows of obstacles in the Cartesian space [6].

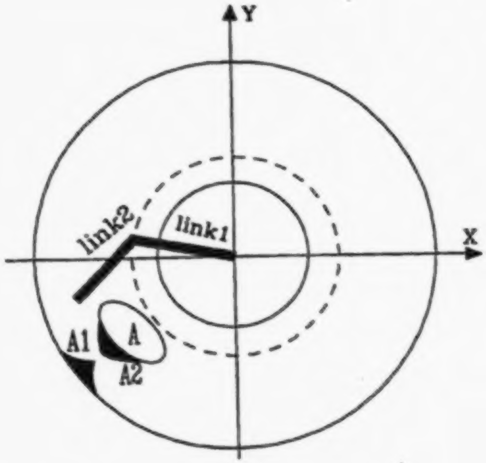


Figure 2 Obstacle and its Shadows

3.2 Strategy

Most of the path planning schemes address the problem of avoiding stationary obstacles in the workspace. Since stationary obstacles always place constraints on the robot motion, it is usually necessary to compute the free space globally. This is often done off-line. However, for the moving objects, they only impose constraints on the robot motion temporarily, i.e. they are only avoided by the robot at a specific point in time and space. In the following, stationary obstacles and moving obstacles are treated differently. Forbidden regions caused by stationary obstacles are computed explicitly in the global sense while the corresponding forbidden regions caused by moving obstacles are computed dynamically and locally.

3.3 Forbidden Regions Caused by Stationary Obstacles

In our algorithm, a discretized C-space representation is used for ease of implementation in order to build the Configuration space sequentially. The C-space is discretized by a well chosen resolution (according to the accuracy and speed of the algorithm) along the θ_1-axis and the θ_2-axis respectively (θ_1,θ_2 are the joint parameters of the robot). Thus the C-space can be considered as a matrix. Each element of this matrix actually represents a certain configuration of the robot. We call this matrix the C-space matrix and it will be denoted as CM.

Let c_{ij} be the element of CM located on the intersection of the *i*-th row and the *j*-th column. We define

$$c_{ij}= \begin{cases} 0 & \text{if the configuration is collision-free} \\ 1 & \text{otherwise (C-space obstacle point)} \end{cases}$$

For a robot with two revolute links, the corresponding CM matrix can be constructed as follows:

(1) Initialization: let c_{ij}=0, for all *i* and *j*.

(2) Ignore link 2. Find the ranges of the forbidden values of θ_1 by considering the rotation of link 1 from its lower-constraint angle to its upper-constraint angle around the fixed base and by taking account of obstacles in the workspace. Fill one column of CM according to the corresponding forbidden ranges.

(3) Sample the legal range (collision free range) of θ_1 at a specific resolution. Find the ranges of the forbidden values of θ_2 by considering obstacles in the workspace and the rotation of link 2 around the position of joint 2 which is determined by the current value of θ_1. Store these forbidden ranges into the CM matrix.

(4) Repeat step (3) for the legal values that correspond to the collision-free ranges of θ_1.

The key step here is to determine the range of the forbidden values of one joint parameter by giving the range of values for the previous joint parameter. For the obstacle modelled by a polygon, Fig.3 illustrates the case when the position of joint 2 is fixed at point P_0.

Suppose that the obstacle is a convex polygon with 5 vertices P_1, P_2, P_3, P_4 and P_5. S_1 and S_2 are the two intersection points between the obstacle and the arc segment caused by the motion of the end-effector of the robot. Let $\alpha_1=\angle AP_0P_1$ be the angle between the line segment P_0A and the line segment P_0P_1 and let $\beta_1=\angle AP_0S_1$ be the angle between the line segment P_0A and the line segment P_0S_1. Assume that γ_1, γ_2 are the real forbidden values (i.e. the values representing the extremities of the forbidden region). For any obstacle within the range of joint 2 at the specific value of joint

1, two corresponding forbidden values are computed as

(a) Find two critical values α_{max} and α_{min}. Here α_{max}=max(α_1, α_2, ..., α_5) and α_{min}=min(α_1, α_2, ..., α_5). Let P_{max}, P_{min} correspond to the vertex with α_{max} and α_{min} respectively.

(b) Calculate the intersection points S1 and S2 between the arc segment and the given polygon and then compute the corresponding angles β1 and β2 with β1 ≥ β2.

(c) Compute the distances between P_0 and P_{max}, P_{min}. Then we can use the following rules to obtain the desired real forbidden values.

(i) If $| P_0 - P_{max} | \leq l_2$ then $\gamma_1 = \alpha_{max}$
else $\gamma_1 = \beta_1$

(ii) If $| P_0 - P_{min} | \leq l_2$ then $\gamma_2 = \alpha_{min}$
else $\gamma_2 = \beta_2$

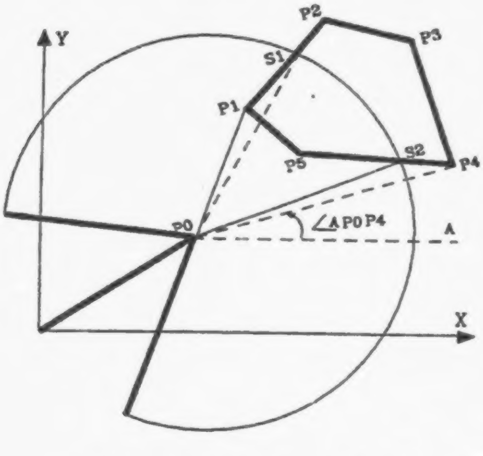
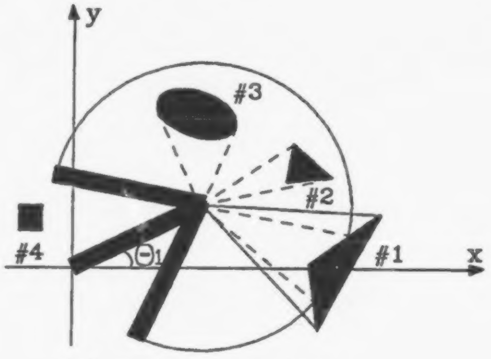


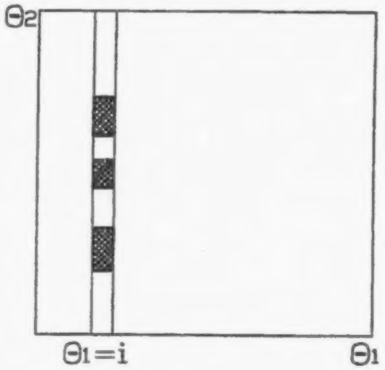
Figure 3 Real Forbidden Values

In Fig.3, it is not hard to see that $\alpha_{max}=\angle AP_0P_1$, $\alpha_{min}=\angle AP_0P_4$, $\beta_1=\angle AP_0S_1$ and $\beta_2=\angle AP_0S_2$. Since $|P_{max}-P_0|$ is less than the length of link 2 and $|P_{min}-P_0|$ is greater than the length of the link 2, therefore the real forbidden values will be given by $\gamma_1 = \alpha_{max}$ and $\gamma_2=\beta_2$ according to the rules discussed above.

In step (2) we actually compute the values for each element in one row of CM and in step (3) we obtain the values for each element in one column of CM each time. Obviously, in step(3), we do not need to consider every obstacle within the workspace. Only obstacles within the specified circle resulting in the rotation of link 2 may cause collision. In order to implement this strategy, the distance between any obstacle and joint 2 is calculated to check for the possibility of any collision. Let D_{2j} denote the distance between joint 2 and the *j*-th obstacle. At stage *k*, only obstacles with $D_{2j} \leq l_2$ have to be considered. This will cause a substantial reduction in the computational burden compared to the commonly used sweeping technique [1][3]. Fig.4(a) shows the situation when link 1 is fixed at θ_1. The corresponding representations of these forbidden regions in the C-space, i.e. one column of CM, is shown in Fig.4(b).



(a) Cartesian space



(b) Configuration space

Figure 4 Obstacles and their Forbidden regions caused by link 2

The same strategy can be applied to link 1. Actually we only need to run step (2) once. If link 1 is blocked in the vicinity of a range of angles, then the corresponding elements of the last row (one may choose any row) of CM will be set to the value of 1. In step (3), the corresponding stages with $c_{mj}=1$ will not be executed. The corresponding forbidden region in the C-space is a straight strip.

3.4 Forbidden Region Caused by Moving Obstacles

When only time-invariant obstacles exist in the workspace, the collision-free space simply depends on the geometry of the obstacles and the manipulator. Since stationary obstacles always impose constraints on the motion of the robot, it is necessary to compute the corresponding forbidden regions globally for the purpose of path planning. On the other hand, for time-varying obstacles, we only need to compute its corresponding forbidden regions locally, i.e. at the specific time interval and the specific space. This is based on the fact that time-varying obstacles impose constraints on the motion of the robot temporarily, i.e. at some specific time and space. When there exist time-varying obstacles within the workspace, the collision-free motion planning for the robot is usually produced step by step due to the time-varying structure of the CM matrix [2]. This requires the implementation of forbidden region computation in real time. Let Δt be the step size during which interval the collision-free motion will be produced for the robot. It is obvious that if we consider the area swept by a moving obstacle during the time interval $[t_k, t_k+\delta t]$ at stage k as a stationary obstacle and if the corresponding forbidden region is computed by employing the algorithm developed for the stationary obstacle, then we will have the presentation of the free space during the time interval $[t_k, t_k+\delta t]$.

The next issue is how to compute the forbidden region caused by a moving obstacle during the time interval $[t_k, t_k+\delta t]$ quickly and efficiently. Since we know the locations of the moving obstacles at time t_k and time $t_k+\delta t$, it is not difficult to determine the area swept by the moving obstacle during the interval $[t_k, t_k+\delta t]$. Also this area can be modeled by a polygon. By employing the same algorithm for the stationary obstacle, the corresponding forbidden region can be easily obtained.

One fact should be noted that at time t_k, the moving range of the robot during $[t_k,t_k+\Delta t]$ is limited by its physical constraints such as the angular velocity and acceleration. Let $\Delta\theta_1$ and $\Delta\theta_2$ be maximum possible moving angles for link 1 and link 2 during the time interval $[t_k,t_k+\Delta t]$ respectively. The possible moving angles for joint 1 and joint 2 are given as follows:

$$\begin{aligned} &\theta_1 \in [\theta_{1k}-\Delta\theta_1, \theta_{1k}+\Delta\theta_1] \\ &\theta_2 \in [\theta_{2k}-\Delta\theta_2, \theta_{2k}+\Delta\theta_2] \end{aligned} \qquad (8)$$

Obviously, we only need to compute the forbidden regions caused by the moving obstacles within the above given ranges. According to the strategy developed above, we only need to calculate the two corresponding forbidden values for each moving obstacle if it is within the robot moving range during the time interval $[t_k,t_k+\Delta t]$. This makes it possible to compute the forbidden regions caused by the moving obstacles in real time.

How can we determine the value of δt? It is obvious that δt should be chosen to be greater than Δt in order to guarantee that within the time period $[t_k,t_k+\Delta t]$ the robot does not collide with any moving obstacles. Since the trajectory of the moving obstacle is independent of the motion of the robot, therefore, the motion produced for the robot during the time interval $[t_k,t_k+\Delta t]$ should also be guaranteed in order to have no collision with any moving obstacles when they are moving from time $t_k+\Delta t$ to time $t_k+2\Delta t$.

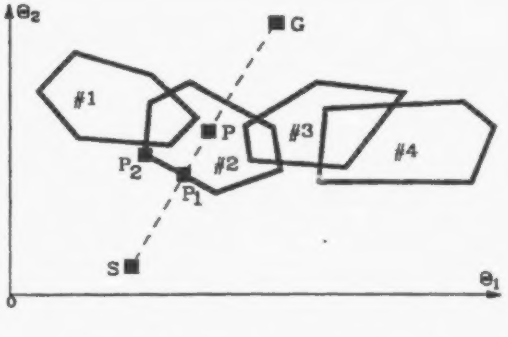


Figure 5 Choice of the time interval δt

This can be shown in Fig.5. Suppose that the C-space obstacles #1, #2, #3 and #4 are caused due to the motion of the moving obstacle during the time periods $[t_k,t_k+\delta t]$, $[t_k,t_k+2\delta t]$, $[t_k,t_k+3\delta t]$ and $[t_k,t_k+4\delta t]$ in the C-space, respectively. Assume that at time t_k the robot is at point s. If we only consider the area swept by the moving obstacle during the period $[t_k,t_k+\delta t]$ as the forbidden region and if we produce the motion for the robot during the period $[t_k,t_k+\Delta t]$, the robot may move directly to point p according to its dynamics. It can be seen that this may possibly cause collision when the moving obstacle moves ahead during the period $[t_k+\delta t, t_k+2\delta t]$. On the other hand, if we consider the area swept by the moving obstacle during the period $[t_k, t_k+2\Delta t]$ as a forbidden region, then the robot can only move to point p_1. Obviously, the bigger the δ is, the safer the path is. However, if δt is selected to be too large, then a substantial part of the free space will be treated as a forbidden region. Therefore care should be exercised in selecting the minimum value of δt in view of the required accuracy. In our algorithm, we choose δt to be equal to $2\Delta t$.

4. C-space Obstacle Approximation

The next step is to address the approximation of the forbidden regions, i.e. the regions with $c_{ij}=1$. We use a bounding convex hull as the approximation of a forbidden region. The computation complexity can be greatly reduced by choosing a good representation of the boundary of the forbidden region and by using some geometric properties.

Note that forbidden regions are calculated by finding two forbidden values for each obstacle within the specific robot moving range at each step for the angle θ2 that correspond to joint 2 while increasing the angle θ1 of joint 1 from its "right limit value" step by step until its "left limit value" is reached (except the block range, i.e. $c_{1j}=1$ for all j). Two data links (e.g. linked data structure) are used to store these two forbidden values where the maximum of the forbidden values is placed in one data link denoted as UL (upper data link) and the minimum of the forbidden values is placed in the other data link denoted as LL (lower data link). The resulting two links will represent the whole boundary of a forbidden region. Since a convex hull in the plane consists of ordered extreme points for the given set of points [8], based on the information stored in the above two data links, we can construct the corresponding convex hull for any forbidden region. The LL will give rise to a polygon chain (lower-hull) which is monotone with respect to the θ_1-axis whereas the UL can give rise to an analogous chain (upper-hull). The concatenation of these two chains is the required convex hull, i.e. the C-space obstacle. The procedure to construct the lower-hull is as follows:

(1) Let the first element of the LL be the origin of the polar system.

(2) Search the LL starting from the successor of the "origin element" to the rear of the link and find the element P which has the maximum angle value (point P is called an extreme point).

(3) Delete all elements starting from the successor of the origin element to the element preceding P.

(4) Let P be the origin of the polar system. If P is the rear of the LL then go to (5), otherwise go to (2).

(5) Stop. The remaining link is the lower-hull.

Similarly, this procedure can be applied to construct the upper-hull. The only difference is that when we search the UL, the next extreme point should have the minimum angle value instead of the maximum angle value in the lower-hull case.

A conventional convex hull algorithm usually requires two steps: (1) identify all extreme points and (2) order these extreme points so that they form a convex hull. Due to the fact that the given set of points is arbitrary and that there is no relation between the points, the algorithm should take each point in the given set of points into account. This will take a considerable amount of time and will need a lot of memory. In our case, it is intuitive that all extreme points are located on the boundaries of the forbidden regions. Therefore, only these extreme points will be found to form a convex hull. This will substantially reduce the computation complexity.

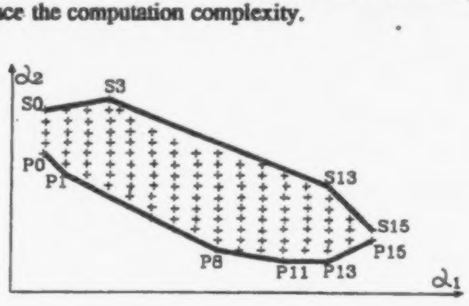
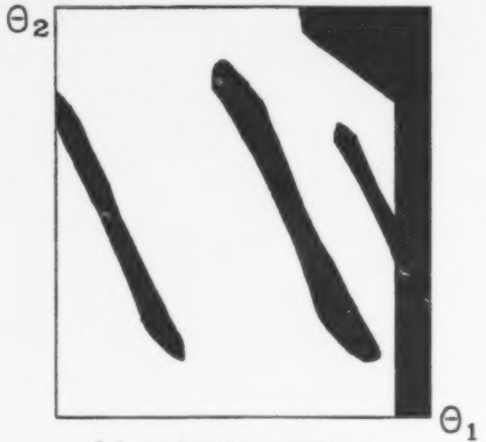


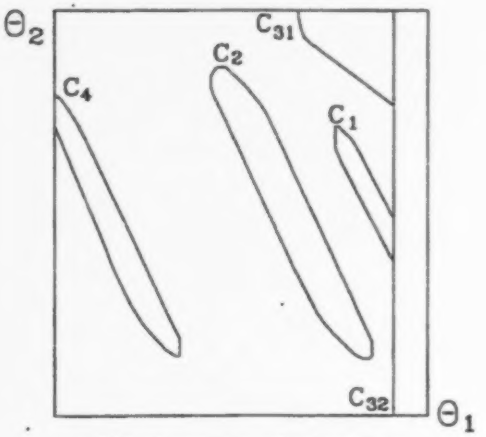
Figure 6 Construction of the C-space Obstacle

Fig.6 illustrates the construction of a C-space obstacle. P_0, P_1, P_8, P_{11}, P_{13} and P_{15} gives rise to the lower-hull (LH). The upper-hull (UH) is composed of points S_0, S_3, S_{13} and S_{15}. The catenation of the upper-hull and the lower-hull gives rise to the corresponding C-space obstacle.

For the working environment shown in Fig.1, the corresponding forbidden regions caused by the stationary obstacles and their approximations are shown in Fig.7(a) and Fig.7(b), respectively. We can see that C-space obstacles C_{31} and C_{32} are caused by obstacle #3. In our algorithm, they are considered as two C-space obstacles. This makes our scheme more accurate than the schemes used in papers [1][3].



(a) forbidden regions



(b) C−space obstacles

Figure 7 Forbidden Regions and their Approximations

5 Conclusion and Discussion

In this paper, a geometry-based fast algorithm for C-space obstacle computation is developed. Obstacles could be either stationary or time-varying. Based on the fact that stationary obstacles always place constraints on the robot motion, while time-varying obstacles only impose constraints on the robot motion temporarily (i.e. at specific time and space), the stationary obstacles and the moving obstacles are treated differently. Forbidden regions caused by stationary obstacles are computed explicitly in the global sense while the corresponding forbidden regions caused by moving obstacles are obtained dynamically and locally. The complexity of the C-space obstacle computation can substantially be reduced by introducing some strategies which are based on the use of some geometric properties. This makes it possible to compute forbidden regions caused by obstacles within the workspace in real time.

In our algorithm, distances among the specific joint and all obstacles are calculated in order to determine if the specific obstacle is within the moving range for some joints. Usually it is very time-consuming to compute these distances. For the sake of real time implementation, we can have some range sensors fixed on the joints of the robot to measure these distances. Also, the algorithm is developed for a planar robot with two revolute joints. However, the technique can be directly extended for the robot with more than two degrees of freedom.

References

1. T. Lozano-Perez, "A simple motion-planning algorithm for general robot manipulators", *IEEE J. Robotics and Automat.*, Vol.RA-3, no.3, pp.224-238 June 1987.
2. C.S. Zhao and M.M. Bayoumi, "Cooperative Motion Planning for Two Robots Sharing a Common Workspace in the Presence of Obstacles", *ICARCV'92 Intern. Conf. on Automat., Robotics and Com. Vision*, RO-9.1.1-5, Sept. 1992
3. K.K. Gupta, "Fast Collision Avoidance for Manipulator Arms: A Sequential Search Strategy", *IEEE Trans. on Robotics and Automation*, Vol.6, no.5, pp.522-532, Oct. 1990
4. O.Katib, "Real-time obstacle avoidance for manipulators and mobile robots", *The International J. of Robotics Research*, Vol.5, no.1, pp.90-98, 1986
5. W.S. Newman and M.S. Branicky, "Real-Time Configuration Space Transforms for Obstacle Avoidance", *The International J. of Robotics and Research*, Vol.10, no.6, Dec. 1991
6. Y.C. Chen and M. Vidyasagar, "Some Qualitative Results on the Collision-free Joint Space of a Planar n-DOF Linkage", *The International Journal of Robotics Research*, pp.1623-1630, 1987
7. V.J. Lumelsky, "Effect of Robot Kinematics on Motion Planning in Unknown Environment", *In Proc. 24th Conf. on Decision and Control*, pp.338-343, 1985
8. F.P.Preparata and M.I.Shamos, Computational Geometry, Springer-verlag, 1985
9. M. Brady, ect. Robot Motion: Planning and Control, The MIT Press, 1982

903104

Development of an Enhanced Mobility Testbed

Alan W. McCormac[1], Chris Brosinsky[2], Doug M. Hanna[3], & Dennis R. Fenrick[4]
Defence Research Establishment Suffield
Box 4000, Medicine Hat, Alberta T1A 8K6

#149743

Abstract

This paper describes a novel vehicle platform under construction at DRES which is designed to exhibit greatly enhanced mobility by employing hybrid running gear. The paper discusses the Articulated Navigation Testbed (ANT) and the considerations which determined its configuration and capabilities. First, as strictly an unmanned vehicle, it must employ teleoperation or some level of autonomous control. Second, it is to possess superior off-road mobility when compared to more conventional vehicles. Increased mobility reduces the computational workload needed to operate in a highly unstructured off-road environment. Third, it must be adaptable and have a flexible mission architecture. By maximizing modularity of the vehicle systems, the design team believes the resultant vehicle will provide a generically mobile chassis upon which DRES researchers can take robotics out of the lab and into the field.

Introduction

In the battlefield of the future, it is very likely that many roles will be filled by unmanned vehicles. The goal of the Defence Technologies Division (DTD) at Defence Research Establishment Suffield (DRES) is to conduct research into creating, controlling, operating, and applying unmanned vehicles in simulated or real military scenarios. The Vehicle Concepts Group within DTD is primarily concerned with research into superior vehicular platforms that facilitate unmanned operation. This paper discusses the design of such a vehicle, which is now being built at DRES. It is important to note that this document is only intended to disclose the design, and not to discuss applications of the vehicle in either manned or unmanned missions.

A review of the literature revealed that many research institutions around the world are developing conformal platforms [2,3,4,5]. These are vehicles which possess special geometry of one kind or another that allows them to conform to highly unstructured terrain. The two most popular formats seen are the doubly-articulated six-wheel-drive rover projects and the hexapod or six-legged walking machine. Both have advantages and disadvantages, evident in the performance of research prototypes.

In an effort to combine the advantages of rolling and walking, DRES personnel conceived the Articulated Navigation Testbed (ANT), a multi-unit, articulated, proof-of-concept (POC) demonstrator. Its locomotion is provided by hybrid running gear comprised of wheels attached to legs. The goal is to produce a "generically" mobile vehicle chassis that facilitates unmanned control and possesses an architecture adaptable to any assigned mission.

Mobility Considerations

Definition of Mobility

Vehicle mobility over land is determined by several factors, many of which specifically relate to vehicle configuration and capabilities. It is very easy to single out one outstanding aspect of the performance of a vehicle and call it a highly mobile vehicle, but it is also very misleading to do so. Researchers from the NATO countries have arrived at a standard for quantifying overall land vehicle mobility in terms of Speed Made Good (SMG), the highest effective speed operating in a prescribed terrain unit of assumed isotropic properties.

Speed Made Good may be calculated using a FORTRAN computer simulation program called the NATO Reference Mobility Model (NRMM) [1]. The NRMM main module predicts the theoretical

---

1 Defence Scientist, Vehicle Concepts Group
2 Mechanical Technologist, Vehicle Concepts Group
3 Head / Vehicle Concepts Group
4 HF Automotive Performance Research, Inc.

limit of speed caused by each of a list of factors, then selects the lowest of this set of speeds as the effective SMG in each terrain unit. If one or more of the limits is zero, then the terrain unit in question is flagged as NO-GO and the vehicle is said to have no trafficability in that unit. In other words, terrain with those properties is impassable for the selected vehicle.

The factors in the NRMM which limit vehicle speed are therefore either controlling factors [C] or NO-GO factors [N], depending on whether the terrain unit is or is not trafficable, respectively. These factors are listed in Table I, along with brief explanations of the vehicle, terrain or scenario parameter which sets each. The table also contains the vehicle modification(s) most likely to improve each limit.

To create a truly mobile land vehicle the greatest possible balance of performance in all areas must be sought. Only then can mobility be considered "generic".

For example, a vehicle capable of climbing a dry 70% grade might be called mobile by the casual observer, but gradability is only one mobilty factor. If the same vehicle loses traction on flat ground in 5 cm of mud it is not a "mobile" vehicle. Collective measures of mobility include, but are not limited to: agility, stability, maneuverability, gradability, trafficability, ride quality, and linear feature crossing capability.

It is possible to identify a few unconventional vehicle technologies which would allow a vehicle to modify some of its characteristics, identified in Table I, in order to enhance its mobility. That is the basis for the highly conformal nature of the ANT. For example, the table indicates that off-road mobility may be enhanced by raising the ground clearance. It also indicates that lowering the center of gravity may aid on-road mobility. The ANT can raise or lower both on demand, unlike most vehicles.

Table I: NRMM Speed Made Good Limiting Factors

Factor Description	Type	Vehicle Characteristic
Excessive sinkage/belly drag	N	Increase ground clearance
Excessive obstacle density	N	Tighten turning, increase clearance
Excessive vegetation density	N	Tighten turning radius, articulate body
Obstacle override hang-up condition	N	Increase ground clearance
Inadequate forward obstacle clearance	N	Increase footing approach angle
Inadequate traction for grade	N	Increase footing contact or number
Inadequate visibility to progress	N	
Ride quality limited (human occupant)	C	Improve suspension dynamics
Tire speed limited	C	Select more suitable tire for mission
Soil/slope/vegetation drag (motion resistance)	C	Increase ground clearance or power
Limited visibility	C	
Maneuvering around obstacles	C	Tighten turning, increase clearance
Maneuvering around vegetaion	C	Tighten turning radius, articulate body
Obstacle impact speed limit	C	Improve suspension dynamics
Obstacle override force (human occupant)	C	Improve suspension dynamics
Driver prudence in vegetation	C	
Skidding due to curvature	C	Increase footing and/or decrease roll
Tipping due to curvature	C	Lower center of gravity

Present Technologies

Present vehicular locomotion systems are not good at all aspects of mobility, and vehicle design always involves compromise in the selection of running gear, defined here as the vehicle subsystem which interfaces to the terrain. All vehicles using conventional running gear suffer in one or more mobility factors. For example, wheeled vehicles are typically more agile and maneuverable than tracked vehicles, but possess higher ground pressures and are therefore less trafficable. Tracked vehicles have lower ground pressure and superior traction and are therefore more trafficable, but are not agile or efficient due to (usually) larger overall mass and much larger internal motion resistance.

Both wheels and tracks have been proven successful for negotiating roadways and moderately unstructured off-road terrains. Neither has exhibited strength in highly unstructured terrain. Very rugged, "mobility-hostile" environments are the targeted application of walking vehicles which are being investigated by several institutions around the world. Some research prototypes employ "pogo-stick" locomotion on anywhere from one to four legs [2,3,4], but others contend that walking locomotion can only be achieved efficiently and stably with six or more legs [5]. Movable legs provide a vehicle chassis with highly variable footing positions, thus allowing it to conform to extremely rugged terrain which would completely block the passage of either wheels or tracks of comparable dimensions.

The Adaptive Suspension Vehicle project in the United States is the most technologically advanced hexapod in the opinion of the authors, and is cabable of effective and stable locomotion by stepping in turn with two intersecting tripod sets (fore and aft legs on one side in conjunction with center leg opposite) [5]. The ASV can negotiate random terrain with discrete elevation changes up to a meter in height as long as sufficient flat spots are available to place three footholds on each step. Unfortunately, due to the complexity of coordinating multipedal stepping motion, vehicles like the ASV are very slow in less hostile terrains, and are easily surpassed by wheeled and tracked vehicles.

The Articulated Navigation Testbed

The ANT will possess superior off-road mobility compared to conventional vehicles. The concept vehicle is comprised of multiple body modules, and is steered by articulation. Each module contains a rigid structure to which member systems of the module are attached, two legs connected to the structure by rotatable hip joints, and wheels attached to the lower ends of each of the legs. Each wheel is mounted to the output shaft of a powered wheel motor.

The vehicle moves by either turning the wheels, rotating the legs to which the wheels are attached, or a combination of both. The typical configuration is a three module platform, although it may be extended to four, five or longer depending on specific mission requirements. It can also be minimally configured with two modules as it is shown in Figure 1, though in this mode, its walking capabilities are diminished.

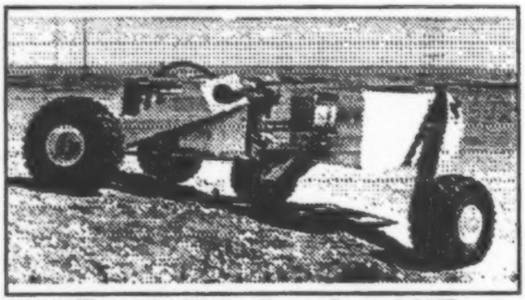


Figure 1: Minimally configured ANT

The minimal configuration would be used in situations where only wheeled operation was required by a mission. However, if a three module platform were used, only two modules (4 wheels) or less are required for locomotion. The third module may then have one or both wheels and/or legs removed, and the wheel motors and/or hip actuators may be used to mount and manipulate auxiliary implements. This modularity was a prime consideration in the vehicle design, so that the third module and any successive modules contain only their required actuators and serve as a skeleton to which mission-specific hardware can be attached.

The articulation joint between modules may have one or two degrees of freedom. Yaw capability is always present, and is moderated by a rotary actuator. The joint may be free in the roll axis, fixed in the roll axis, or controlled in the roll axis through the inclusion of a rotary spring, damper, and/or, actuator. Pitch freedom or compliance is not required, as one principal motivation for using legs is the ability to follow longitudinally variable elevations with a relatively flat body geometry.

The vehicle employs multi-body articulation, so each chassis module follows the arc travelled by the preceding module, and may exactly follow the track of the preceding module if leg geometry is selected appropriately. Thus in terrain with high obstacle density, and particularly in vegetation, the articulated steering provides successive obstacle avoidance for all trailing modules.

Modular Architecture Considerations

Powertrain Modularity

In order to accommodate the many degrees of freedom of the vehicle, power delivery is accomplished by an electrical, hydraulic or combined electro-hydraulic distribution system, depending upon the selection of actuators required for the vehicle mission. In keeping with the design goal of modularity, and in order to provide a slight rearward balance to the vehicle, the rear module shown in Figure 2, is reserved for power generation equipment. It contains an internal combustion engine used solely to drive one or more hydraulic pumps and/or electric alternators, depending once more on the type of actuators used.

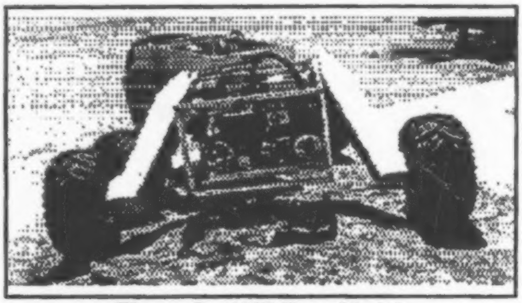


Figure 2: *ANT Rear Module*

Power is distributed through a central manifold in each chassis module, and bussed between modules in paired cables and/or fluid hoses. The second module contains its own manifold and actuators, and also houses the batteries, primary fuel cell for the engine, and the central computer system which coordinates the vehicle motions.

One principal advantage with electric or hydraulic wheel drive systems is a simplified powertrain control function. Mechanical transmissions use clutches and gear sets which add weight and are quite inflexible. Speed is controlled by varying engine throttle. Electric or hydraulic wheel drives provide gearing, both forward and reverse, speed control, and braking through fluid flow or current in a flexible cable or hose, respectively. Vehicle speed and direction are integrated into a single control function, which is likely non-linear or piecewise linear at best, but may be linearized by software.

Control System Modularity

The many degrees of freedom in this vehicle make it capable of very complex motion; and some degree of computer control is essential. It would take a sizable team of human operators to coordinate all the actuators visually. Complementing the modular chassis design, a downward delegation of control responsibility is designed into this vehicle. Each leg/wheel combination is managed by a microcontroller using feedback from transducers on the hip actuator and wheel motor. All microcontrollers are connected in a network mastered by a central processor. In the event a leg is replaced with an auxiliary implement as mentioned earlier, the microcontroller associated with that leg is also replaced with a microcontroller programmed to operate the added implement.

The principal advantage to this distributed control approach is that it largely simplifies the central processor task load. Periodically, the central processor downloads new set points for the actuators, and the closed loop control algorithms executing in the microcontrollers run continually based on the current set point. The accuracy of control is enhanced by this approach because the distributed controllers can run at update frequencies of several hundred samples per second due to the small computational requirements they have. Large scale chassis geometry changes require a relatively slow dynamic response, and are not even required during travel on smooth terrain. This means the central processor does not have to run a continuous control algortihm. The slaved controllers take care of all the real time feedback control loops.

A secondary advantage to the distributed control approach is enhanced self-recovery. In a military scenario, it is quite possible that one or more legs may be lost or crippled. While such an occurrance will certainly affect the overall mobility of the vehicle, the central processor will not be directly affected by the lack of response from the damaged appendage(s). The vehicle will retain some degree of "limp-home" capability as long as the engine continues to operate.

The central processor in the ANT is not necessarily a computer system. It may in fact be a human operator employing joysticks or potentiometers to establish controller setpoints. It may also be a very complex computer system capable of navigating the vehicle autonomously over rugged terrain. It is important to reiterate that the vehicle design team set a goal of achieving generic mobility, not specific to the future applications of this platform in either manned or unmanned missions.

Proof-of-Concept Platform

Simulation

DRES employs DADS (Dynamic Analysis and Design System) for modelling vehicles and vehicle subsystems [6]. It is a comprehensive 3D rigid body simulation package complete with mechanical, hydraulic, and control element libraries, an X-windows interface, and a utility program which extracts model elements from Pro/Engineer, a parametric CAD/solid modeller also in use at DRES.

Kinematic model

A kinematic model was constructed of a three-unit ANT, to be used as a Proof-of-Concept (POC) design aid. It contained three identical chassis bodies and six leg/wheel bodies with the relative motions of joints being prescribed by drivers. Kinematic models do not include dynamic response characteristics, mechanical flexure or control constraints of course, but rather make the assumption the hip angles and wheel speeds are achieved instantaneously.

Dynamic model

The current simulation model of the ANT is dynamic, including force and torque response data for the actuators and motors, tire flexure, and full computation of the equations of motion based on mass and inertia properties of the bodies. This model is used to determine the capabilities of the vehicle when constrained by those parameters.

The dynamic model will be validated by the Proof-of-Concept vehicle, and when the simulated performance becomes a suitably accurate representation of the vehicle performance, the model may be used for two follow-on tasks: optimization of the vehicle design for any future revisions, and the simulation of ANT vehicles in operational scenarios.

Construction

Status

The Proof-of-Concept vehicle is nearly complete mechanically, with all powertrain components and actuators in place. Considerable effort was expended in placing feedback sensors for all actuators and motors in protected locations, while ensuring each modular subsystem remained together.

Hardware and software development has advanced using off-chassis electronics where possible, as the networked microcontrollers are not yet installed. Development of the microcontroller network software is on the critical path to the eventual fielding of the ANT vehicle.

When the networked distributed controllers are mounted and operational, the present system integration phase will be complete. It is expected that shakedown trials will begin early in 1994. During the summer, a validation study of the dynamic model will be conducted.

Plans

The POC ANT platform will be retained by DRES and serve as a technology demonstrator. One possible application in the near term would see the ANT in service as a tow vehicle for a mine detector platform known as the Vehicle Mounted Ordnance Detector (VMOD). Normally pulled behind a truck, the VMOD platform could become a front-mounted mission module of the ANT, enhancing the vehicle safety, because detection would occur before the vehicle passed over the mine, and the vehicle could avoid the mine by lifting the appropriate wheel.

The dynamic model and lessons learned from the POC vehicle may be used to create one or more application variants to suit Canadian Forces or Canadian industry. Areas of application may include, but are not limited to: mine detection and/or clearing, reconnaisance, surveillance and target acquistition functions. The vehicle could also be applied for remote observation of dangerous areas in mining, or for selective logging operations in remote areas in forestry.

Application to higher research areas include the possibility of using connectionist control architectures to investigate self-modifying behavior based on sensing of the local environment. Can the ANT be made to detect and avoid obstacles, with little or no human interaction? Can it be made to identify friend or foe and take appropriate action? These are some of the questions the authors intended this technology demonstrator to answer. Integration of the ANT with other DRES research activities in robotics is planned to follow the model validation phase.

Summary

The principal objective in creating the Articulated Navigation Testbed is to demonstrate a platform with "generic" mobility. It is to combine the efficient mobility that conventional platforms have on roadways and mild cross-country terrains with the enhanced mobility that conformal platforms have in very rugged cross-country terrains. The inherent modularity in the vehicle presents many possibilities for application.

The Articulated Navigation Testbed has been simulated and is under construction at a Proof-of-Concept stage. If Canadian Forces or Canadian industrial concerns have identifiable requirements for a similar vehicle, DRES will be in a position to assist. Further technology base development will be aimed at supporting the continuing research thrusts of DTD unmanned vehicle research.

References

1. Ahlvin, R.B. and Haley, P.W., NATC Reference Mobility Model Edition II, NRMM-II User's Guide, Dept. of the Army, US Army Corps of Engineers, Washington D.C., December 1992.

2. M'Closkey, R.T. and Burdick, J.W., "Periodic Motions of a Hopping Robot with Vertical and Forward Motion", International Journal of Robotics Research, Vol 12:3, June 1993.

3. Shih, C., Gruver, W.A. and Lee, T., "Inverse Kinematics and Inverse Dynamics for Control of a Biped Walking Machine", Journal of Robotic Systems, Vol 10:4, June 1993.

4. Digney, B.L. and Gupta, M.M., "A Distributed Adaptive Control System for a Quadruped Mobile Robot", IEEE ICNN '93 Conference Proceedings, San Francisco, March 1993.

5. Kwak, S.H. and McGhee, R.B., "Rule-Based Motion Coordination For The Adaptive Suspension Vehicle On Ternary-Type Terrain", Dept. of Computer Science, Naval Postgraduate School, Monterey, California, December 1990.

6. "Dynamic Analysis and Design System Users Guide", Computer Aided Design Software, Inc., Box 203, Oakdale, Iowa, 1991.

905C20

Beyond the Space Age...
Environmental Robotics: Nuclear Site Remediation

Martin Brooks
Head, Knowledge Systems Laboratory,
Institute for Information Technology, National Research Council of Canada,
Building M-50, Montreal Rd., Ottawa, Ontario
K1A 0R6

#149745

Abstract

The US Department of Energy has published a 5 year plan for clean up of sites involved in US nuclear weapons production. This plan features robotics as a key technology. The challenges inherent in these applications will drive both robotics business and robotics research in the coming years.

Canadian companies are playing a key role in this growing market, including RSI Research (Sydney BC), Spar Aerospace (Brampton ON), International Submarine Engineering (Port Coquitlam BC), and Robotech (Calgary AB).

The National Research Council is supporting Canadian companies desiring to enter this market in two ways: (1) technology development, including range sensors and mobile robot systems; and (2) coordinated demonstrations, where multiple partners show their technology in large scale demonstrations.

Editor's Note: If you are interested in obtaining this paper, please contact the author at the above address.

900367

An Architecture for Adding Intelligence in a Multiprocessor Environment

Ed Suen
CAL Corporation,
1050 Morrison Drive, Ottawa, Ontario, K2H 8K7

Robert Hayashi & Ann Dent
Defence Research Establishment Atlantic,
PO Box 1012, Dartmouth, Nova Scotia, B2Y 3Z7

#149746

Abstract

This paper describes a multiprocessor software architecture used for the evolutionary development of an Artificially Intelligent Assistant Classification System (AIACS) for interpreting passive sonar data. This architecture supports the fine-grained integration of the two major AIACS functions located on separate processors, an expert inference engine and a user interface for extracting vessel classification signature features from the sonar data.

The layered AIACS architecture is built on an object-oriented programming foundation. Its major elements are the Model-View-Controller (MVC) framework and the Blackboard model. The latter provides the framework for the implementation of knowledge-based problem solving. The former introduces an extended form of object-oriented message passing which dramatically reduces the coupling between event-driven software modules. MVC is extended to a multiprocessor environment via the Remote Procedure call. The development of AIACS serves as the context for the discussion of the architecture.

The potential generalization of this architecture as the basis of a software bus for distributed systems is discussed. The bus is simpler and more suited to expert system development than emerging standards for object-based distributed computing.

Introduction

Expert system technology provides a means of introducing intelligence to problem solving by computer processing. The Artificially Intelligent Assistant Classification System (AIACS) is an example of the use of this approach by the Defence Research Establishment Atlantic (DREA) to develop in collaboration with industry, an interactive expert system solution to a very difficult military problem, the identification of classifications of vessels from opportunistically acquired acoustic (sonar) data. The paper focuses on a unique aspect of this concept demonstration expert system, its software architecture. AIACS was designed and implemented based on a layered multiprocessor software architecture which provides an abstract structural view of the system software. This architecture, formulated and refined during the project for evolutionary development of the expert classification system, is described with reference to AIACS. The generalization of the architecture as a software bus for distributed systems is also discussed.

Overview of the Artificially Intelligent Assistant Classification System

Expert systems formulate decisions through the management and processing of various types of information. A high level information management and processing representation of the vessel classification system is presented in Fig. 1.

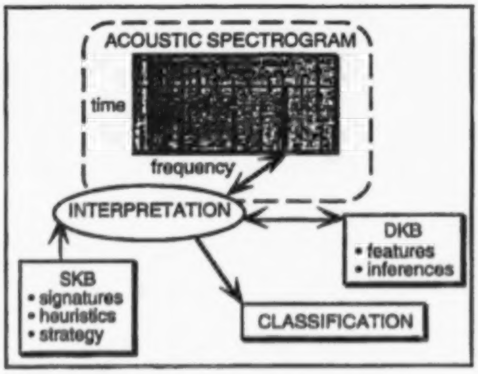


FIGURE 1: Overview of Spectrogram Interpretation.

The processing performed by the expert system, *Interpretation*, is specified in terms of inputs and outputs. The *Acoustic Spectrogram* presents signature data for vessels on a grey-level, time-frequency plot for which pixel intensity represents detected power. The *SK* (Static Knowledge) consists of factual information about signatures of classifications of vessels, heuristic and experiential knowledge, and knowledge of the detailed strategy used in conducting Interpretation. Much of the knowledge for the vessel classification domain is security classified and therefore cannot be reported openly. The *DK* (Dynamic Knowledge) consists of information which is specific to individual examples of the vessel classification problem, such as intermediate inferences generated during problem-solving. A major kind of intermediate inference consists of acoustic features detected on the spectrogram and their recognition as signature features.

Incompleteness and uncertainty of both the acoustic data and the accumulated factual knowledge makes the interpretation problem a particularly challenging application for the development of an expert system. This problem is not solvable by a simple pattern matching approach. AIACS conducts data interpretation by managing and processing acoustic information and domain knowledge according to a modified hypothesis and test strategy. It proactively controls problem solving but throughout the interpretation, it passively monitors and assesses the dynamically unfolding analysis, generates advice and opportunistically exploits any inputs made under operator initiative to advance the solution state.

The following is a list of the domain knowledge essential for implementation of the strategy: descriptions of hypothesis cues and organization of the cues in a decision tree, specifications of acoustic signatures for vessel classification organized in a specialization-generalization hierarchy, and heuristics for signature feature recognition and exploitation [Dent 90, Hayashi 90]. Each cue maps into a hypothesis set of vessel classifications. Recognition of generic cues leads the system to move along the decision tree by seeking more specific cues interactively. In general, smaller hypothesis sets are associated with more specific cues. The last successfully recognized cue provides a set of potential classification identifications and drives the problem solving into the test phase. Confirmation can be conducted as a knowledge-driven search for signature features based on the known signatures of hypothesized vessel classifications. Heuristic knowledge provides the basis for making decisions about which classification should be tested first and what constitutes a successful confirmation. In cases of confirmation failure, conflict resolution strategies allow the system to minimize backtracking. The changing constraints during problem solving can be managed using a truth maintenance facility.

Software Development Environment

A multiprocessor computer, the Macivory, was used as the hardware platform for the development of AIACS. The Macivory provides both a Macintosh IIfx and a Symbolics LISP Machine environment. Although each environment has its own operating system and support services, the Macintosh input/output (I/O) facility serves both processors. The processors share the I/O resource by communicating with the Remote Procedure Call (RPC) facility [Symbolics 90].

The adoption of the Macivory as the hardware for the evolutionary development of the expert system was influenced by practical considerations. At the outset of the development, resources available and the state of symbolic computing technology were significant factors. The initial studies on the exploration of assistance concepts for the vessel classification problem were conducted on Symbolics computers. However, the Symbolics computer lacked a grey-level display capability. It was realized early that the ultimate practical success of a concept demonstration system for the vessel classification problem would depend to a great extent on the development of a user friendly operator interface to support the interactive detection and extraction of acoustic features from spectrograms. The Macintosh provided a well established and widely recognized multiple monitor, Graphical User Interface technology. This was attractive for the implementation of an electronic metaphor of the physical acoustic analysis environment. The Symbolics provided a capability for rapid prototyping of expert system software, based on a powerful, high speed, specialized symbolic computing processor.

The dual processor environment allowed specialized classification functions to be supported on the more suitable processor. Fig. 1 indicates how the system was partitioned over the two processors. The shading denotes the portion implemented on the Macintosh. The major system component on the Macintosh is the graphical interface through which the expert system acquires signature features from a high resolution spectrogram display with interactive support from the operator. The application software on the Macintosh is coded using object-oriented programming based on Think C. The LISP Machine supports the symbolic computation of the expert system. The LISP software is also object-oriented, but based on Flavors. An expert system tool kit, Joshua, provided rule and object representation of knowledge and a truth maintenance facility.

Evolutionary Development Approach

AIACS requires interdependent expert components to conduct feature extraction and to generate classification inferences from features.

Many practical expert systems leave the low level task of input information abstraction to the system user in order to reduce the complexity of system implementation. Such an approach was adopted for the development of AIACS. Consequently, an operator interface was provided to facilitate interactive extraction of acoustic features.

An evolutionary approach was taken to the development of the expert system. The operator interface [Canadian Astronautics Ltd 91] was first implemented as a standalone, two-monitor analysis workstation. The workstation enables an operator to conduct the classification task electronically by: (i) selecting and displaying multiple resolutions of acoustic spectrograms, (ii) overlaying and manipulating graphical icon-based tools on spectrograms in order to recognize signature features by pattern matching, and (iii) extracting features for automatic recording in an appropriate database. An example of a mouse-operated tool, a harmonic set analyzer, can be seen extending across the upper area of the spectrogram shown in Fig. 1. It consists of a horizontal line, a set of short equally spaced vertical lines and two mouse sensitive alpha/numeric icons used by the operator for initiating the creation of data representations of detected harmonic sets. The spacings of the vertical lines can be adjusted in unison using the mouse in order to detect harmonic sets by overlaying a set of (in this case poorly defined) spectral lines.

This workstation was evolved into an interactive interface by incrementally adding knowledge-based functionality to support system initiation of feature extraction through context relevant feedback of SK and DK through the analysis tools [CAL 91]. These knowledge-based analysis tools, provided the system with a capability to interactively constrain in a way relevant to the unfolding analysis context, the manual search for signature features by the operator. For example, some tools operated by automatically feeding back static knowledge to the operator about mappings of significant features to classifications.

The interactive interface system was subsequently extended into AIACS by implementing expert system capability for inferring vessel classifications from extracted features [Hayashi 89a,b]. The expert system and the operator interface are tightly coupled to provide a seamless integration of the expert inference capability with the interactive feature extraction capability. The difficulty in developing an intimately coupled system over two distinct, loosely coupled processors was addressed by replacing the database with an object-based Dynamic Knowledge Base (DKB) that was implemented based on the architecture described in the next section. The discussion of the DKB in the context of the architecture, presented later, indicates that the DKB serves as an intermediary through which the user interface implemented on one processor communicates information with the expert system inference engine implemented on the other processor.

The DKB stores all analysis state information for ready access by the expert system component of the augmented system. This provides the inference engine with the necessary information to monitor and reason about the evolving analysis in order to guide operator actions. The DKB was extended to include executable code. Executable code allows the operator interface to directly invoke expert system functions (e.g. asking the expert system to provide directions). Moreover, the executable code supports an asynchronous messaging capability which allows the expert system to automatically control the operator interface in a knowledge-driven fashion [CAL 93].

Architectural Framework

The architectural framework used to develop AIACS incorporates both the Model-View-Controller (MVC) [Krasner 88] and Blackboard [Hayes-Roth 85] frameworks. Fig. 2 shows the layered nature of this architecture.

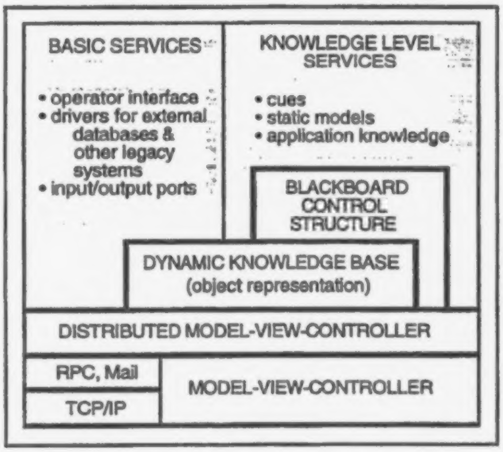

BASIC SERVICES
• operator interface
• drivers for external databases & other legacy systems
• input/output ports
KNOWLEDGE LEVEL SERVICES
• cues
• static models
• application knowledge
BLACKBOARD CONTROL STRUCTURE
DYNAMIC KNOWLEDGE BASE (object representation)
DISTRIBUTED MODEL-VIEW-CONTROLLER
RPC, Mail
TCP/IP
MODEL-VIEW-CONTROLLER

FIGURE 2: Layered AIACS Architectural Framework.

Each architectural element is built on top of a lower supporting element and depends directly on the latter's functionality. For example, the RPC (Remote Procedure Call) layer is constructed on the TCP/IP (Transmission Control Protocol/Internet Protocol) layer [Tanenbaum 88]. The overlying layer integrates its own contribution with that of the under layer to extend the architectural framework for the development of application systems. The interaction interfaces between layers is represented by the thick boundary lines. Most of the architecture is implemented with tools that provide object-oriented programming capability.

Infrastructure-Level Bus Layers

The elements are either layers or plug points. The shaded areas in Fig. 2 denote the two plug points: Basic Services and Knowledge Level Services. The other elements—MVC; TCP/IP; RPC, Mail; Distributed MVC; DKB; and Blackboard Control Structure—are all layers forming the infrastructure part of the architecture. The layers each have a fixed set of external entry points, functions that can be called by external users of the layer. These layers as a group are the basis for the software bus of AIACS. The plug points are where application-specific software components can be plugged into the generic bus infrastructure.

This bus infrastructure is the software analog of a hardware bus [Shelton 93], and carrying the analogy further, the plug points correspond to slots that allow customizing boards to be added to an open architecture system. A software board is a package of software components residing on a single machine. (As an aside, it is noted that the LISP processor is implemented in the Macivory workstation as a coprocessor board plugging into a hardware bus slot of the Macintosh computer.) The generic infrastructure-level layers hide the distributed hardware configuration from the application-specific components. This greatly simplifies the development and maintenance of distributed systems.

The Model-View-Controller layers

The MVC framework, the foundation of the architecture, was originally developed to build pluggable user interfaces. At a conceptual software design level, MVC introduces a metaphor commonly used for developing interactive graphical applications. It is essentially a technique for partitioning software functionality that ensures the following three key criteria are met for an MVC compliant software module: (1) the module is event driven; (2) messages are broadcasted indirectly to other MVC modules through some set-like data structure; and (3) the number of core communication primitives is small.

At the code implementation level, MVC introduces into the architecture, an additional layer of dynamic binding of methods to messages [Ellis 90]. The extra dynamic binding capability is achieved by broadcasting messages through a set-like structure of recorded dependents, rather than directly to the dependents themselves. The implementation uses the normal dynamic binding capability of an object-oriented language to ensure that the handler for a message is dependent on the type of the object. In addition, since the sender always broadcasts a message through a set-like structure, a procedure call may result in an unknown number of messages.

One of the object-oriented programming languages used for the development of AIACS supported the implementation of MVC through a generic class library [Think Class Library 91]. Since the class library code was not available for the other processor, counterparts of the key classes required to support MVC were implemented on the other processor.

The implementation of a spectrogram analysis tool can be used to illustrate the application of the MVC framework. The code for an analysis tool is implemented by means of three essentially independent modules. The three components of the Model-View-Controller name indicate the individual functionality of the software modules. The *model* module contains and manipulates the state and control information [Canadian Astronautics Ltd 91]. The *view* module implements the display of the tool on a spectrogram. The *controller* module translates low level I/O events (operator input actions) to higher domain-specific actions for the model. Each module can be implemented by a suitably defined object (involving appropriately specified methods). The controller and view modules can be combined when implementing a tool for which behavior is directly linked to appearance. When the model changes because of operator input, update changes are broadcast to a list of its dependents which contain its view and controller. A point to note is that the model does not directly address the dependents so it does not require explicit assumptions about even the presence or the number of dependents. The view and controller can then ask the model about its new state and update the display and behavior respectively. The view and controller can be modified independently of the model so the look and feel of a tool can be easily modified.

In the general case, an MVC compliant software component communicates indirectly with others through change announcement messages sent through a set-like data structure that records component linkages. The initiator makes no direct reference to external components. The receiver components are individually responsible for reacting appropriately to the change announcement messages.

A tool is invoked by creating an instance of the tool's model object. The design of a tool using the MVC framework provides the convenience to create tools that can be operated in several ways. The following are examples of complex tool operations that have been implemented in the interface system. A tool can be displayed and manipulated simultaneously on spectrograms of different resolutions. This was done using different view objects to handle the separate displays for the multiple view tool. Two copies of a tool, one in single view and another in multiple view mode, can both be present on the same spectrogram. The same code is used for both configurations. Separate instances of the tool model are created and different linking configurations are established on-the-fly at run time.

The AIACS architecture extends MVC for application to a distributed environment by combining the MVC framework with the RPC mechanism. RPC, in conjunction with its underlying TCP/IP layer, provides the inter-processor communication functionality for calling procedures from one processor for execution by another processor and for receiving the results of the execution. The Distributed MVC capability is illustrated by Fig. 3. Four processors P1, P2, P3, and P4 are shown. Change announcement messages flow from one object (O1 or O2, etc.) to another object. The flow (represented by the arrows) is via RPC (and potentially mail) for objects located on different processors. The bi-directional arrow indicates the general case in which an announcement message triggers a return message. Incidentally, P3, considered in isolation (without the arrows between its O1 and P2's O1 and O2) depicts a single processor MVC. Also a set of its three objects (e.g. O2, O4, and O5) that communicate with each other in pairs can represent the software implementation of a MVC-compliant analysis tool residing completely on processor P3.

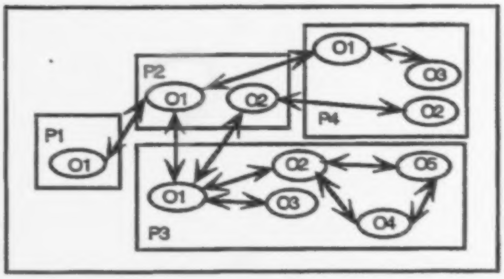


Figure 3: Distributed Model-View-Controller.

It was indicated earlier that the indirect change announcement communication protocol of MVC supports the dynamic coupling and decoupling of software modules. This dynamic coupling between software modules supports a great deal of flexibility and reconfigurability, as in the single processor case. Consequently, distributed software components can be interchangeable and pluggable. This is the basis of the flexibility of the AIACS architecture.

The software reconfigurability provided by the Distributed MVC layer was exploited in adding expert system capability to the user interface system. The model modules of the MVC-compliant tools were moved to the LISP processor to consolidate all the analysis state information. This was done by writing some local RPC code which allows client-server communication to be implemented as procedure calls instead of I/O commands. It was not necessary to re-implement the software for the tools.

The DKB layer

The DKB is constructed on top of the Distributed MVC layer. The structured DKB is the AIACS implementation of the global blackboard described in [Corkill 87]. This DKB structure supports the Blackboard Control Structure described below. Software components plugged into the bus as MVC-compliant objects are typically inserted into the DKB. The object may act as either a service provider or client. The DKB provides transparent, unified access to objects distributed over multiple processors.

Clients of the DKB are shielded from the physical multiprocessor distribution of the lower-level object system. The DKB object therefore may be located on either the Macintosh or LISP processor or even both. DKB query mechanisms support retrieval of objects corresponding to arbitrary selection criteria. For example, in AIACS, knowledge sources which are components of the Knowledge Level Services use the query mechanisms to assess the overall solution state. In addition, the interactive interface which is plugged into the Basic Services queries the DKB in order to provide the operator with a view of the measurement objects [CAL 91]. The global state information provided by the DKB is critical to command and control decision support functions.

The use of a central DKB structure and the asynchronous MVC messaging protocol enabled the two tightly coupled AIACS subsystems to be developed in an isolated, stand-alone manner. It has only recently been observed that these mechanisms can be generalized as a basis for a software bus for distributed systems.

Blackboard Control Structure

The Blackboard Control Structure layer [Corkill 87] is used by the Knowledge Level Services to control multiple sources of knowledge. The Blackboard Control Structure ensures that appropriate knowledge sources receive the required state information in an event-driven manner. With this support, inferences may be made in both a reactive and directed manner. In AIACS, the control structure software interacts with appropriate knowledge sources to manage and process information to conduct data interpretation by hypothesis and test. In a more general application of the architecture, this interaction supports hierarchical command and control in distributed systems.

Services Layer

There are two types of plug interfaces, MVC and non-MVC. Non-MVC software components can only be clients of bus services. Since non-MVC components cannot be tightly coupled with the bus, this type of plug interface is not discussed any further.

The Basic Services plug point supports application-specific operator interfaces and drivers for various I/O ports to databases or other legacy systems. Legacy systems may be introduced in a black box fashion in this plug point. The legacy system functions as a sink or source of data when it is interfaced with a driver (an MVC-compliant object plugged into the bus).

The Knowledge Level Services plug point contains the processing smarts of the system. In particular, command and control decision making components are contained in this layer. For AIACS, the application-specific components are cues, signatures and rule-based signature state models. Under the Blackboard architecture terminology, cues and signature objects are knowledge sources. They contain both domain knowledge and control knowledge. Control knowledge is represented as demon procedures and methods attached to these objects. Sets of interacting knowledge source objects provide the hypothesis generation and hypothesis confirmation services. For example, cue objects interact to support hypothesis generation. This plug point may also contain language mechanisms supporting the electronic representation of command and control knowledge [Hayashi 92]. In AIACS, these language mechanisms are used to build cues supporting reactive behavior (for monitoring) and state models (for predictive forecasting) [Dent 90, Hayashi 92].

General Application of the Architecture as a Distributed Systems Bus

The architecture facilitated the evolutionary development of AIACS. A user interface implemented on a single processor was extended into a multiprocessor expert system through reconfiguration of software rather than replacement. It also simplified the implementation of the software for interactively conducting complex data interpretation.

The paper now focuses on the potential for more general application of the multiprocessor architecture that was originally used to develop an application-specific system. AIACS components, distributed over two processors, collaborate to conduct data interpretation. A more general case of collaborative problem solving would involve a large system consisting of many distributed component systems, including expert systems like AIACS, that each conduct specialized problem solving.

There are two ways in which distributed systems conduct cooperative problem solving. The familiar one, data oriented computing, involves indirect interactions among processes through use of shared data stores. Data integrity is maintained by transaction management facilities and this helps to ensure cooperative behavior among the multiple processes [Gray 93]. Data oriented computing cannot be scaled up to support time critical systems containing autonomous, continuously operating, dedicated computing devices. The more general one, process oriented computing, supports direct interactions among processes. (At the software implementation level, direct interactions between processes are translated to procedure calls.) Although the restrictions of communicating via shared data stores are removed, shared data stores may still exist, either within the processes, or as sinks or sources external to the system.

Mechanisms derived from the Distributed MVC and the blackboard elements of the architecture—the many-to-many, real-time linking for executing related procedures, and the declarative representation with its event-driven Blackboard Control Structure supporting the management of procedures—are the keys to the AIACS bus solution for process oriented computing. These mechanisms, in conjunction with the plug points and the reduced object coupling enforced by MVC, make it possible to build a simpler and more flexible generic bus infrastructure for collaborative problem solving by distributed application-specific objects.

The MVC approach brings benefits that flow from: avoiding direct source code reference between MVC components, supporting asynchronous messaging and providing a scalable public interface. Since MVC is applied uniformly to lower level components, more detailed interfaces can be easily constructed to these subordinate components. This option is not available for procedural systems residing inside an MVC wrapper. The ability to provide a more detailed interface is likely to be important for expert system applications where one component controls and monitors the functions of another. It is likely that fine-grained interfaces will be evolved during the iterative development of the expert system.

The declarative representation feature of the blackboard structure is the key to managing process oriented collaborative problem solving. Distributed blackboard systems have been used in Artificial Intelligence research for several years as a declarative state representation technique supporting central command and distributed control in expert systems [Corkill 87]. The benefits of the blackboard framework are transferable to the command and control of general purpose distributed systems.

The empirical exploration of the architecture's potential for use as a generic software bus for distributed systems was not within the scope of the AIACS project. Therefore this possibility is discussed by reference to new technologies for data oriented and process oriented distributed computing. Architectural lessons will be drawn from the unique aspects of the AIACS implementation.

The AIACS architecture is quite similar in many ways to emerging industry tools and standards [Miller 93]. The Object Exchange in Borland's Workgroup Computing Architecture [Borland 93] is similar in many respects to the DKB. Like the DKB, the Object Exchange contains all shared objects. The publish-and-subscribe model of the Object Exchange is similar to the DKB object insertion and MVC dependency links of AIACS. Objects are published in the Object Exchange; objects are inserted into the DKB. Clients subscribe to objects in the Object Exchange; clients register themselves as dependents or providers in AIACS.

The extra layer of RPC indirection supported by Distributed Object Management System (DSOM) [Udell 93], which isolates components from distribution issues, is similar to the distribution transparency supported by the Distributed MVC layer in AIACS. DSOM software components communicate with each other within a single processor environment using the Workstation Runtime engine which may be transparently substituted to support object communication across system boundaries. As in AIACS, software components can be constructed without worrying about the location of fellow collaborators.

The AIACS architecture is unique in its pervasive use of the object-oriented MVC framework for system components. The extra layer of indirection supported by the MVC set data structures isolates the components from distribution issues. Underlying bus transport mechanisms deliver the change announcements messages from object to object, possibly across system boundaries, as shown in Fig. 3. The underlying bus transport mechanism can be a custom implementation as in AIACS, or any one of the newer industry standards. Consequently, MVC is quite a suitable technique for structuring software in anticipation of emerging industry standards.

The Object Management Group's Common Object Request Broker Architecture (CORBA) bus places no such constraints (MVC compliant) on its client components. Consequently, CORBA does not support asynchronous messaging. This is considered to a be a major drawback [Shelton 93] since real-time applications require this capability. In contrast, MVC-based client objects inherently support asynchronous messaging. Furthermore, asynchronous messaging is essential for a blackboard control structure. A blackboard control structure supports the implementation of the agents and wizards that are likely to be common in a groupware environment.

The frame-based persistent object store used by the Newton personal digital assistant [Halfhill 93] is similar to the frame-based DKB of AIACS. The use of symbolic slot data structures to represent services provides a number of benefits. This declarative representation supports a query interface such as that found in Object Linking and Embedding 2.0 (OLE 2.0) [Williams 93]. With this interface, objects can determine the availability of a method, instead of blindly sending a non-supported message. The externally visible state representation of the DKB also supports a blackboard control structure.

In summary, the software boards should be object-oriented and MVC-based from the ground up. Even if the underlying distributed systems bus does not support asynchronous messaging, the boards, themselves, should support this capability.

Over time, industry standard tools will replace the Distributed MVC and DKB layers of the AIACS architecture. The remaining MVC layer supports integration into the industry standard and the blackboard control structure layer supports agents, wizards, and management tools. In the interim period before the establishment of industry object-oriented distributed computing standards, the benefits of distributed computing can be achieved with the direct use of the Distributed MVC and DKB layers of the AIACS architecture. These layers may even be necessary after this interim period to integrate specialized processors or older computer systems not capable of supporting new tools.

Summary Remarks

AIACS was produced through evolutionary development by augmenting an operator interface system with expert system capabilities on a multiprocessor computer. An MVC-based architecture was devised and used for the design and implementation of the interactive interface system on a Macintosh processor. It was refined into a layered architecture which simplified the development of a knowledge-based system which seamlessly integrates the interface system with an expert inference capability, implemented on a Symbolics

LISP processor. The refined architecture is based on a Distributed MVC framework, developed by combining the single processor MVC paradigm with the RPC mechanism and a blackboard framework.

The architecture used for developing the multiprocessor AIACS can be generalized to serve as the basis for a software bus for distributed systems. The bus would support the integration of MVC-compliant plug-in software, thereby enabling system functionality to be modified incrementally. The software bus addresses many problems associated with complex distributed systems, issues concerned with integration, centralized command and distributed control, and system extension. A distributed systems technology that is based on a modular software bus would support cycles of evolutionary acquisition of computer-based systems by a military organization, facilitate integration of commercial off-the-shelf systems, and improve the performance of distributed data interpretation applications while lowering development and maintenance costs.

Some architectural lessons can be drawn from the AIACS implementation. MVC is a suitable technique for structuring software components in a distributed environment. In addition, a uniform declarative object representation, in conjunction with MVC, supports a blackboard control structure that is desirable in a distributed environment. Simple distributed computing buses can be constructed for existing operating systems, if the components are MVC compliant.

References

Borland Workgroup, [1993] Computing Vision: A White Paper, Borland International, Sept. 4, 1993.

CAL Corporation, FirstMark Technologies Ltd, I.M.P. Group Ltd, [1991] "Software Design and Implementation for an Intelligent Assistant Classification System", Contract W7707-0-1297/01-OSC, DREA CR 91/441, July 1991.

CAL Corporation, FirstMark Technologies Ltd, [1993] Final Report for Contract W7707-0-1795/01-OSC, in preparation, 1993.

Canadian Astronautics Limited, FirstMark Technologies Ltd, I.M.P. Group Ltd, [1991] "Software Design and Implementation for a Basic Computer Assisted Classification System", Contract W7707-8-1003/01-OSC, DREA CR 91/408, January 1991.

Corkill, D.D., Gallagher, K., Johnson P.M., [1987] "Achieving Flexibility, Efficiency, and Generality in Blackboard Architectures", Proceedings AAAI-87 Sixth National Conference on Artificial Intelligence, pp. 18-23, 1987.

Dent, C. A., [1990] Private Communications, 1990.

Ellis, M.A., and Stroustrup, B., [1990] The Annotated C++ Reference Manual, Addison-Wesley, pp. 208-213, 1990.

Gray, J., and Reuter, A., [1993] Transaction Processing: Concepts and Techniques, Morgan Kaufmann Publishers, 1993.

Halfhill, T.R. [1993], "PDAs Arrive", BYTE, Vol. 18, No. 11, pp. 66-94, October 1993.

Hayashi, R.M., [1989a] "Framework for Evolutionary Development of an Autonomous Expert System for Acoustically Identifying Classifications of Vessels", DREA Technical Memorandum, DREA TM/89/202, 1989.

Hayashi, R.M., [1989b] Informal Communication 1989.

Hayashi, R.M., [1992] Informal Communication 1992.

Hayashi, R.M., [1990] Private Communications, 1990.

Hayes-Roth, B., [1985] "Blackboard Architecture for Control", Artificial Intelligence, 26, pp. 251-321, 1985.

Krasner, G.E., and Pope, S.T., [1988] "A Cookbook for Using the Model-View-Controller User Interface Paradigm in Smalltalk-80", Journal of Object-Oriented Programming, August/September 1988.

Miller, M.J., [1993], "Applications Integration: Making Your Programs Work Together", PC Magazine, Vol. 12, No. 6, pp. 108-137, 30 March 1993.

Shelton, R.E., [1993], "OMG's CORBA 2.0: Industrial-Grade Standard for Distributed Object Computing?", Distributed Computing Monitor, Vol. 8, No. 5, 1993, pp. 3-16, May 1993.

Symbolics [1990] User's Manual. Book 13 Networks, Symbolics Inc., Burlington, MA. 1990

Tanenbaum, A.S., [1988] Computer Networks, Second Edition, Prentice-Hall, Inc., Engelwood Cliffs, N.J., 1988.

Think Class Library V1.1 [1991] Object-Oriented Programming Manual, Symantec Corp., Cupertino CA. 1991.

Udell, J., [1993], "IBM's Assault on Distributed Objects", BYTE, Vol. 18, No. 12, pp. 125- 126, November, 1993.

Williams, S., [1993], "An MSJ Interview with Microsoft's Chief Architect of OLE, Tony Williams", Microsoft Systems Journal, pp. 55-66, October 1993.

404945

A method of interactive recognition of spatially defined model deployment templates using abstraction

Steven Woods
Defense Research Establishment Valcartier,
Combat Intelligence Automation Group,
Courcelette, Quebec G0A 1R0.
woods@jupiter.drev.dnd.ca

#149747

Abstract

Abstraction has been suggested intuitively and shown experimentally to be an effective method of improving the efficiency of various types of problem solving applications such as scheduling and planning [11, 13]. In this paper we present an interactive system for model deployment template recognition that uses abstraction as the basis for both model representation and user interaction, and which provides a convenient model for iteratively improving confidence in proposed template matches or solutions.

The problem

An intelligence analyst must draw on specific experiential and learned knowledge of enemy behaviour in order to recognise specific deployment patterns signifying organisational relationships of enemy elements sighted. One aspect of this complex task is to recognise the existence of enemy "doctrinal templates" which detail the patterns in which military units are commonly deployed for specific activities. These "templates" represent a snapshot view of how the elements of a military grouping are commonly positioned. Since military units are often strictly structured and deployed according to fixed procedures, a fairly accurate "typical" representation can be built of particular formations undertaking certain activities in the field, such as "Motor Rifle Regiment (MRR) in attack". The analyst must fuse spatially and temporally disjoint messages or sightings into a coherent picture of the entire situation, including higher level formations and their activity. The individual elements of the MRR may be envisioned as "slots" in the template, and the rules or restriction regarding deployment of these elements can be represented as constraints amongst the template slots. Sightings are derived from intelligence messages received from a diverse set of sources and include direct sightings and contact reports and electronic or other sensor telemetry. The current set of sightings is referred to as the "situation", and each sighting as a "situation element".

One major difficulty with such a task is that these templates are embedded within an extremely complex, dynamic and incomplete "view" of the enemy. Both the current situational battlefield view and the deployment templates themselves are approximations of the real situation. Messages from the field may contain vague or conflicting composition or location information, and inaccurate messages may be propagated into inaccurate conclusions. Templates defined in terms of spatial relationships between components of specific types, sizes, activities and orientation approximate reality by allowing certain tolerances on the attributes or spatial constraints. Considering a template as a set of "stick-pins" of varying types and sizes inter-connected by a set of rubber-bands with a minimum and maximum stretch capability, the process of template matching reduces to finding appropriate situation elements in which the stick pins may fit and where the bands between the pins allow this fit.

Matching of these templates to incompletely specified situational views can result in many partial solutions where only some subset of the pins could be fit to a situation element. In these cases we may wish to determine which of many partial solutions is more promising, possibly according to predefined criteria or measures. In addition, the knowledge of the precise structure of these templates themselves is only approximate, and the templates themselves may evolve over time. Consequently, any attempt to assist the analyst recognition task with limited automated tools will need to be able to take advantage of the experience and flexibility of an intelligence analyst, and any such system must therefore be designed with this interaction as a primary goal.

There are many interesting research issues in the formulation of spatial deployment templates as well as in the development of an interactive process capable of assisting in template recognition in noisy situations. While we concentrate here on the construction of the process, this work builds upon the representation work in [9] where we introduced one structure capable of capturing analyst's *Expectations* based on reported sightings, and allowing for the explanation and propagation of these into later conclusions about the situation. The interactive approach discussed in this paper is essentially a continuation of our work on modelling analyst's expectations by allowing for the definition of commonly seen or expected patterns of deployment, and of modelling the analyst's process used to recognize these patterns.

This paper has two primary goals. First, we describe the way in which we model templates and use these models to locate template instances in large data sets. Second, we detail the model of interaction and instance identification we have developed, noting the importance of abstraction as an interaction methodology which allows an intelligence analyst to locate instances in a timely manner, provides approximate solutions where complete ones are absent, and which can be tailored to fit specific analyst needs or problem instances.

Template modelling

The intelligence messages containing enemy unit or equipment sighting information are of varying accuracy and completeness with respect to perceived structure, location and time. Messages describe a sighting at some point in time $T1$, are sent for collation at some other time $T2$, and arrive later at $T3$. The analyst attempts to organize hundreds of these messages at some still later time $T4$ into a coherent picture of the world at $T4$. Any sighting or perception of the world at $T1$ decays or loses its information value over time in a dynamic setting where units move, and groupings disintegrate into smaller elements, or reintegrate to form new formations. As a result, analysts commonly attempt to form a picture for some past time TP somewhere before $T4$ and after $T1$ based on all reports available at $T4$. This single-time model created of the past at TP will be considered to be the input to our system. Essentially we are working with a blurry snapshot of a moment in the past. While the ultimate goal of this research is to attempt to recognize templates according to a dynamic model over time, in this paper we discuss a static model only.

Templates defined as groupings of constrained element slots may be represented by a constraint graph, where the nodes of the graph represent the slots, and the arcs between nodes represent constraints. Constraints between slots typically but not exclusively capture spatial relationships, and reflexive constraints for each node may identifying features such as type, size, or orientation. Problems specified in this fashion, where a solution is composed of a set of consistent element-to-slot assignments are well described in [4], and are known as a Constraint Satisfaction Problems (CSPs). An example of a template constraint graph is shown in Figure 1 with five node slots where each slot is constrained by several attribute ranges, and 11 inter-slot spatial constraints on distance, position, and orientation.

We are interested in finding all possible instances of a particular template in a set of situation elements, and consequently we are interested in finding all solutions to a CSP modelling our template recognition problem. We have constructed a non-interactive system known as the MOdel Template Object Recognition System (MOTORS) [10] which is capable of solving CSPs in general, and spatially oriented CSPs such as template recognition in particular.

In the literature, CSPs are typically solved via some search strategy coupled with domain-dependent heuristics, via some graph-reduction method which attempts to propagate consistency throughout the problem graph, or through some hybrid approach utilizing both search and propagation. Recent experimental results [7] have shown that hybrids which perform either some degree of constraint propagation *before* or *during* search tend to have better performance than either individual approach. It has been shown that search strategies with "look-ahead" properties can be thought of as augmenting search with some degree of during-search constraint propagation. By defining search strategies in this way the amount of constraint propagation adopted during search can be more easily identified and adapted, and various strategies more easily compared empirically. Although [7] indicates which type of hybrid strategy is best for some small toy domains, published results solving interesting problems using this strategy are difficult to find. The doctrinal template recognition problem provides an excellent example of a moderately complex problem which can be addressed in a reasonable length of time utilizing hybrid search.

In our work with MOTORS detailed further in [10], we experiment at some length with combinations of search, propagation algorithms, and various heuristics. For instance, template recognition is a domain that is inherently spatial, and while the spatial relationships may be precisely represented in the form of inter-slot constraints, a template-wide constraint that requires that all situation elements in consideration for a particular instance exist within a certain "range of focus" has been adapted in MOTORS as an example of a specific domain-dependent heuristic for template recognition. The interactive model discussed in this paper is designed so as to provide a framework in which further research in appropriate combinations of heuristic search and propagation can be undertaken.

Figure 2 details a situation with one embedded Figure 1 template instance amid 300 randomly created and positioned elements. The time required by a simple backtracking configuration of non-interactive MO-

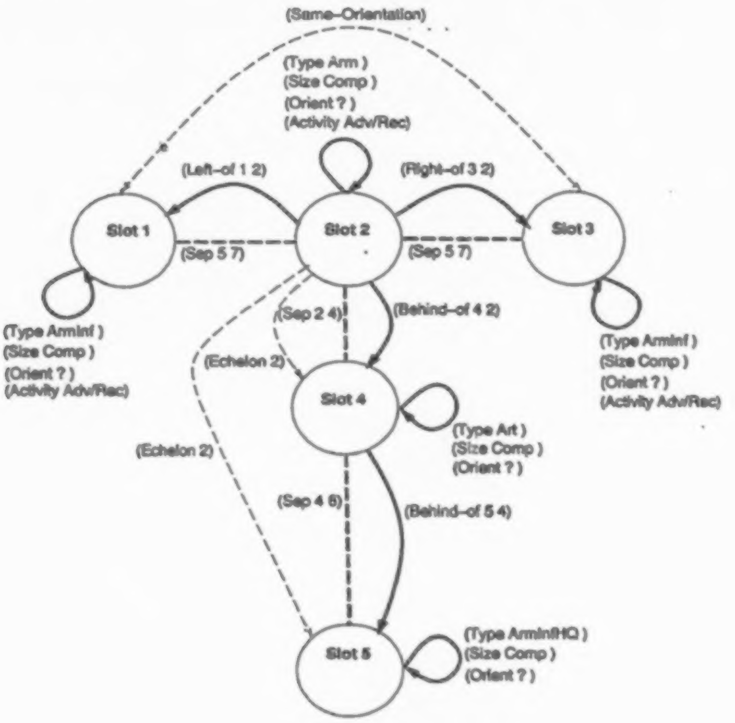

(Same–Orientation)
(Type Arm)
(Size Comp)
(Orient ?)
(Activity Adv/Rec)
(Left–of 1 2)
(Right–of 3 2)
Slot 1
Slot 2
Slot 3
(Sep 5 7)
(Sep 5 7)
(Type ArmInf)
(Size Comp)
(Orient ?)
(Activity Adv/Rec)
(Sep 2 4)
(Behind–of 4 2)
(Echelon 2)
(Type ArmInf)
(Size Comp)
(Orient ?)
(Activity Adv/Rec)
Slot 4
(Type Art)
(Size Comp)
(Orient ?)
(Echelon 2)
(Sep 4 6)
(Behind–of 5 4)
Slot 5
(Type ArmInfHQ)
(Size Comp)
(Orient ?)

Figure 1: Whole CSP 5 variables, 11 arc constraints.

TORS parameters to find all instances (in this case one) of the template in this situation on a Sparc10 workstation is less than 1 second. Figure 3 shows the template instance in its same orientation as in the example situation, along with two instances that only partially match the template slots. It is quite easy to see how difficult the process of identifying these templates can be amidst noisy data. The manner in which inter-slot spatial relationships can flex greatly complicates the identification process for the human observer, and template recognition can be seen as much more complex than identifying objects with regular structure and orientation.

Experimentation and research involving hybrid search using domain-independent heuristics for selection of propagation methods, methods of sharing constraint propagation work during search, and domain-dependent spatial heuristics are discussed further in [10].

Abstraction and partial solutions

In light of our preliminary results which showed that moderately complex situations could be solved completely for one template instance with relatively little difficulty, we considered how to better model the partial matches that must be accommodated in a real situation. Completely matching a particular template involves matching the template's slots with specific enemy elements such that the all spatial and identity constraints between slots are satisfied. When such a complete assignment cannot be found within the element set or situation, it may be acceptable that some subset of slot assignments that only satisfy some "sufficient" portion of the constraints be accepted. This partial solution to the problem is a candidate which may suggest the existence of unsighted situation elements, and the analyst must accept or reject these based on past experience. Advance analyst work in describing acceptable or "plausible enough" partial solution characteristics and identifying template constraints which may be relaxed in the search for usable solutions will limit the partial instances located.

Given a template definition, two approaches of locating partial solutions for CSPs are immediately evident: solving some portion of the original CSP, or solving an abstraction of the original CSP. In the first case, the problem could be partitioned into independent parts, where the lack of a solution to one part would not preclude the solution to another part, and the solved part would represent a partial solution of the whole problem. In the second case, the problem could be considered wholistically where the most "important" or "key" aspects of the whole problem could be identified and solved initially. Solutions for the ab-

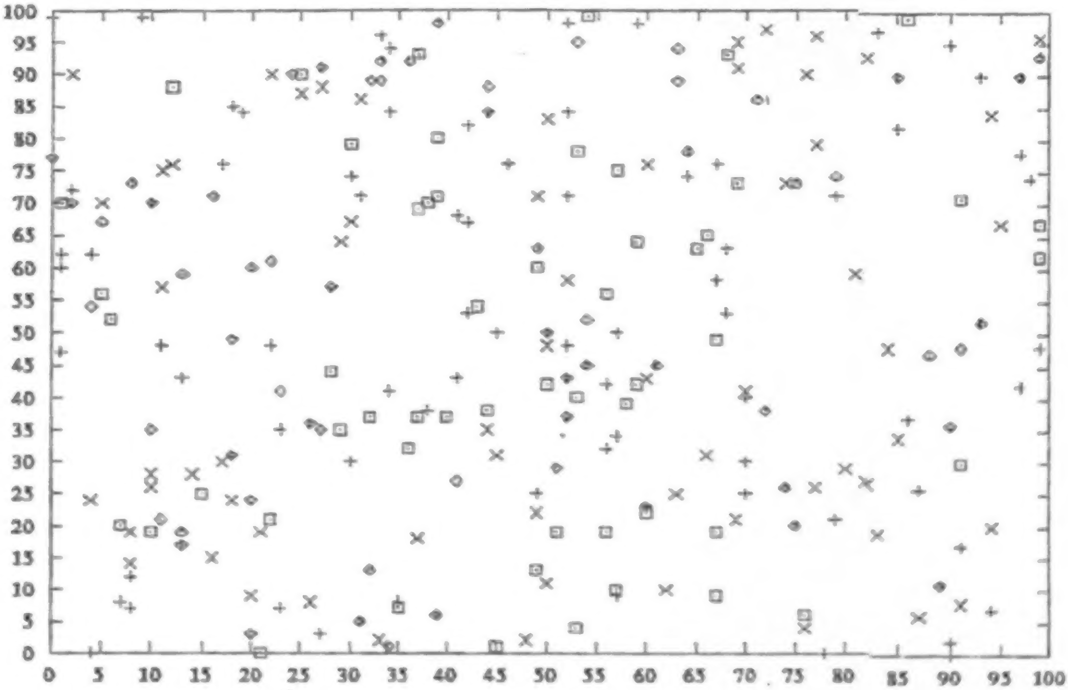


Figure 2: Situation with 300 unit sightings and 1 whole template instance.

stracted problem composed of only "key" constraints and slots could then be refined by progressively fitting problem detail if possible, increasing the perceived degree of fit for those solutions satisfying additional constraints. Intuitively the progressive fit is attractive when considering systems that must offer solutions in domains where the problem structure itself has been defined according to incompletely captured expert knowledge. In [10] we discuss partition of CSPs further, but this paper will concentrate on developing a model for the solution of CSPs based on solving abstract problem representations before more specific or detailed ones.

In [2], the notion of Partial-CSPs (PCSPs) is outlined. A PCSP is a "relaxed" or "less constrained" version of the original CSP. As originally formulated, if a solution could not be found to the fully constrained problem, one could relax the problem in one of several possible ways and attempt to solve this new, simpler problem. This process of attempt, fail, simplify and repeat continues until some "minimally relaxed" CSP is solved. The repeated relaxation of the original CSP defines a space of relaxed CSPs with the original CSP at the root, and the most relaxed CSPs at the leaves, as shown in Figure 5.

The specification of an abstraction hierarchy for a particular CSP involves the identification of the most "important" variables and constraints. Abstraction is often taken to mean specifying the differences between "more specific" and "less specific" problem representations. There are exactly 3 ways of creating an abstracted CSP: removing variables or slots, removing constraints, or relaxing constraints. If the "most abstract" problem representations have the "most im-

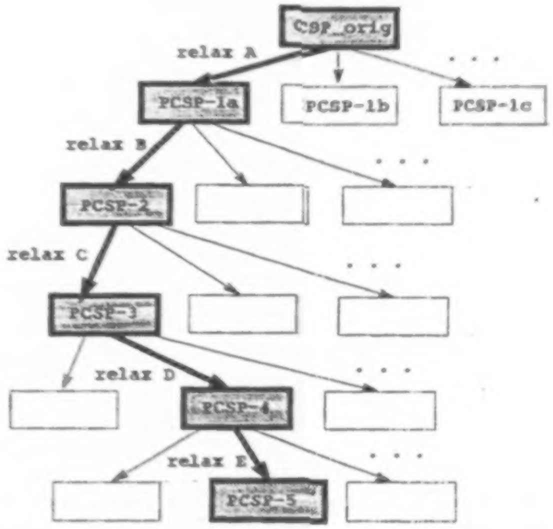


Figure 5: One abstraction hierarchy in a PCSP space.

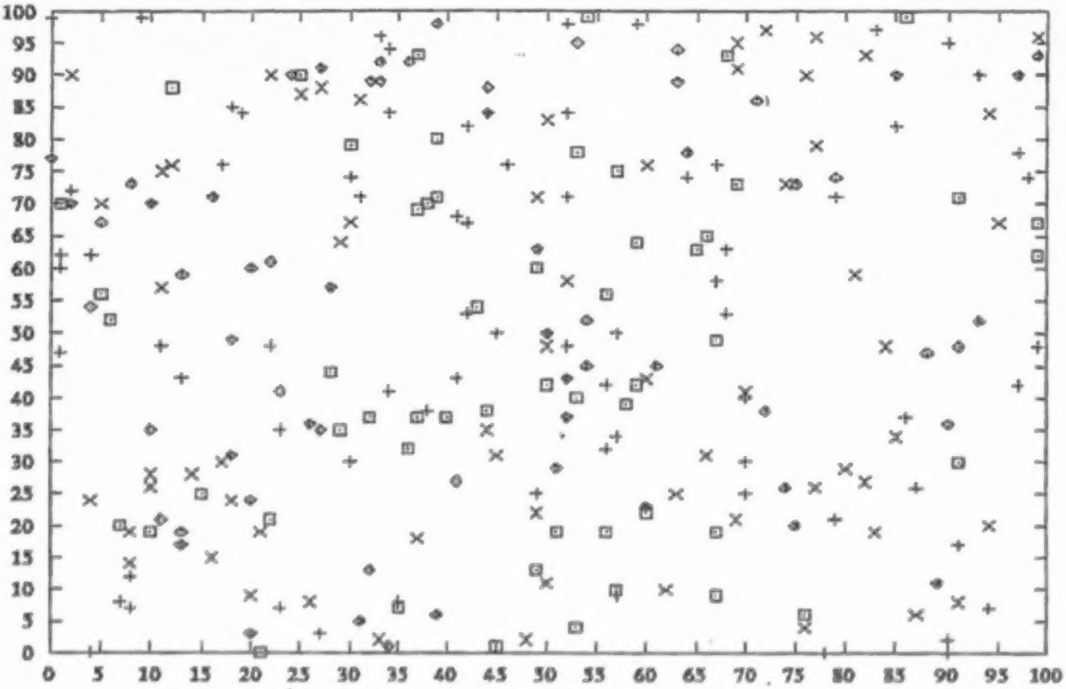


Figure 2: Situation with 300 unit sightings and 1 whole template instance.

stracted problem composed of only "key" constraints and slots could then be refined by progressively fitting problem detail if possible, increasing the perceived degree of fit for those solutions satisfying additional constraints. Intuitively the progressive fit is attractive when considering systems that must offer solutions in domains where the problem structure itself has been defined according to incompletely captured expert knowledge. In [10] we discuss partition of CSPs further, but this paper will concentrate on developing a model for the solution of CSPs based on solving abstract problem representations before more specific or detailed ones.

In [2], the notion of Partial-CSPs (PCSPs) is outlined. A PCSP is a "relaxed" or "less constrained" version of the original CSP. As originally formulated, if a solution could not be found to the fully constrained problem, one could relax the problem in one of several possible ways and attempt to solve this new, simpler problem. This process of attempt, fail, simplify and repeat continues until some "minimally relaxed" CSP is solved. The repeated relaxation of the original CSP defines a space of relaxed CSPs with the original CSP at the root, and the most relaxed CSPs at the leaves, as shown in Figure 5.

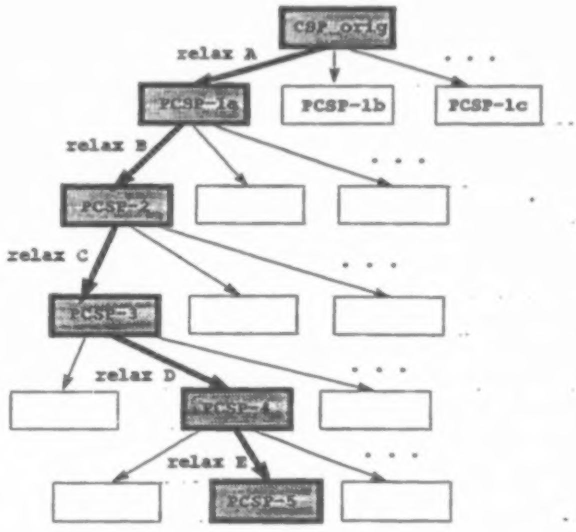


Figure 5: One abstraction hierarchy in a PCSP space.

The specification of an abstraction hierarchy for a particular CSP involves the identification of the most "important" variables and constraints. Abstraction is often taken to mean specifying the differences between "more specific" and "less specific" problem representations. There are exactly 3 ways of creating an abstracted CSP: removing variables or slots, removing constraints, or relaxing constraints. If the "most abstract" problem representations have the "most im-

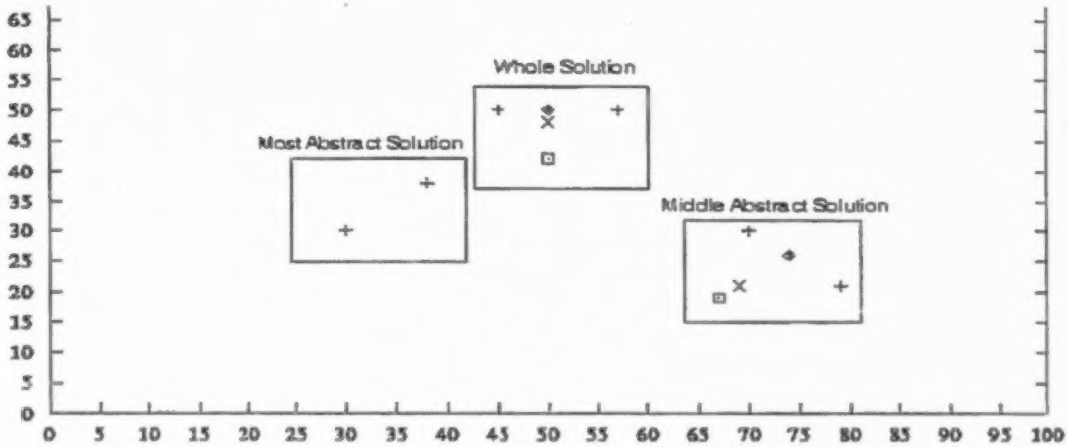


Figure 3: Template instances hidden in Figure 2.

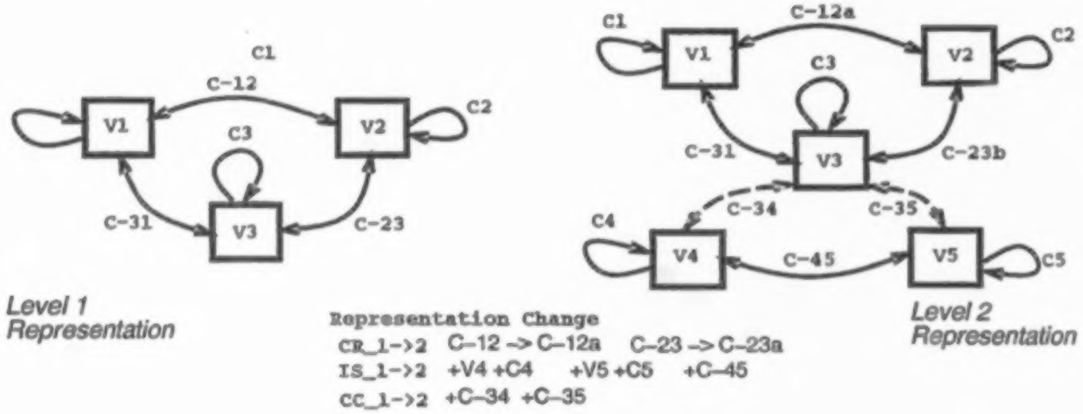


Figure 4: Representation Change between abstraction levels.

portant" nodes and constraints, and less abstract representations have more "detail", then an analyst will have to develop an abstraction hierarchy for the recognition of templates by first identifying a set of the most important or critical template aspects*, and forming one model using only these aspects. Next, the analyst will have to develop one or more subsequent models that successively incorporate more detail. Arranging these models such that the "least detailed" model is considered "most abstract" and the "most detailed" is considered "least abstract" gives rise to an abstraction hierarchy where a solution to the most detailed model can be viewed with the most confidence, while a solution to a less detailed model would have to be viewed with less confidence. If an abstraction hierarchy were created with three levels, we might consider labelling the first level "plausible", the second "supported", and the third "highly supported". Partial solutions within each level, could be ordered further according to other heuristic measures.

As part of the process of creating a hierarchical description of template recognition, the expert analyst must rank critical problem aspects according to importance. These problem aspects that need to be considered are:

1. Ordering Variables by "importance".

   In ordering variables, the intent will be that the abstract representation of a particular problem will include only those variables of most importance. At each successively more specific layer of the abstraction hierarchy, variables would be added to the representation.

2. Ordering Constraints by "importance".

   As for variables, the intent is that an abstract representation of a particular problem includes

---

*One example of military critical element identification that supports this model is *Signature Equipment* [8, 1], defined as "any item of equipment which reveals the type and nature of the unit or formation to which it belongs".

only the most important constraints. Clearly the definition of "important" constraints depends to some extent on which variables are applicable at each level. In fact there are two classes of constraints which differ fundamentally. Node or "reflexive" constraints affect only a single variable and as such may be ordered in terms of importance either with respect to a particular variable, or grouped by constraint type. Arc constraints affect a set of variables and may be categorized with respect to either that set of variables or by constraint type.

3. Relaxing Constraints.
   In addition to constraint ordering, constraints or groups of constraints may themselves have their restrictiveness softened.

After identifying the relative importance of these problem aspects, a problem representation hierarchy must be formed. Work has been done elsewhere towards automatically creating these hierarchies [3], however, here we consider only expert constructed hierarchies. Creating an appropriate hierarchical model is of great importance since the model will completely determine which, if any, candidate template instances will be found. A particular concern for the analyst lies in careful construction of abstract templates, since construction of a template in too strict a fashion will result in desired matches being missed.† Additionally, where we examine only those abstract solutions where key or important elements were identified, it is quite possible that a template containing "key" equipment or patterns has been missed despite its presence as a result of the key equipment not being sighted. In these cases, where some partial template could potentially be found, but where key equipment is missing, we can either fail to include such partial template occurrences or allow for more searching, possibly by using some preference heuristic that only "prefers" partial solutions with key elements over non-key partial solutions. The way in which we define both our abstraction hierarchies and our control strategy will determine the amount of flexibility with which we match templates. One specific method of creating a hierarchy based on importance factors is described in [12].

One immediate benefit of applying abstraction to CSPs is the ability to use hierarchies to represent problems in a way which will suggest a reasonable manner of solving problems in an approximate-first approach where the lack of a complete "least-abstract" representation solution will not leave the user without some "best-guess" near-solutions. Further, these near-solutions are arrived at systematically in that the user has implicitly indicated what constitutes progressively improving confidence for a particular problem domain. If we consider creating a hierarchical model as the process where a complex problem or object is successively simplified in specified ways, we see the creation of one path through the entire PCSP space as described in [12], from original CSP to least specific PCSP on the leaf. Figure 5 shows a PCSP space with a highlighted path showing how PCSP-5 was derived from the root CSP.

The PCSP space reflects all possible relaxations of the root CSP. A path from root to leaf indicates the set of relaxations undertaken to yield a certain PCSP. This path essentially defines an abstraction hierarchy for a particular problem, where the leaf node on the path is the most abstract, and the problem is made successively more detailed as we ascend towards the root. Since each successor in the PCSP space is found by applying one relaxation on its parent, the change in representation of the problem from one level to the next is exactly one operation. If we delineate certain points in the PCSP space as fundamentally changed problem representations, then representation change operators between levels may be formed into groups or sets. An abstraction hierarchy is therefore formed that maintains all of the representation change operators between root and leaf, while each group of these representation change operators fully describes the representation change between one abstract level and the next. Explicit use will be made of these representation change descriptions in our abstract search method, and it is interesting to note that many other abstraction-based search systems [13] fail to take advantage of this explicitly represented domain information, using only information implicit in the descriptions of each abstraction level.

If we consider the example template of Figure 1 as the most detailed or root model of our full PCSP space, we must identify the important or critical aspects as described earlier in order to create a hierarchy for the identification of this template. As an example, we can create an example most abstract version by identifying Slot 1 and Slot 3 as critical identifying features of the whole model. Crucial constraints between these are identified through consultation with the analyst. Additional detail is added to our model at the next abstract level, as Slot 2 and 5 are added, with additional constraints, and restrictions on earlier constraints. Finally, the most detailed model of Figure 1 is created with the addition of Slot 4, constraints connecting Slot 4 to the earlier Slots, and the replacement of earlier constraints with more restrictive ones.

---

† In a related work, Nadel [6] shows clearly that different CSP representations of the same domain have radically different solution characteristics, suggesting that future research is required in order to further guide expert construction of templates.

Search and interaction

We are interested in combining search and constraint propagation in a model that can support user interaction during search. Analysts may possess specific domain knowledge allowing them to remove some candidate partial solutions early in the search process, or identify which are most promising. Other work [12, 5] details approaches to solving CSPs based on "local search" with "repair heuristics" which take an initial assignment of elements to nodes and attempts to repair the instantiations until some acceptable measure is met. This type of approach has been shown to be effective when searching for one or several solutions, however, in the template recognition domain, we may be forced to find all solutions in our situation, and as a result we focus on complete, constructive approaches for the basis for our system.

Search in CSPs is basically a repeating process of selecting a variable to instantiate, selecting a domain value to assign to that variable such that the assignment is consistent with previous assignments, and continuing until all variables are assigned. Each complete assignment of domain values to variables is a solution. Essentially, if a problem P initially has N variables, each subsequent assignment results in a remaining problem P_1 with one less variable than its predecessor. We know that the process of finding all possible solutions is simpler if the domain sizes of each variable are smaller. Application of various constraint propagation techniques with varying effectiveness and cost could conceivably reduce each variable domain. Whether or not the reduced search achieved will exceed the work spent simplifying the problem is difficult to determine in advance, and more research is warranted into characterizing situations where this propagation work is justified, and in determining which propagation methods which would most effectively be applied.

Breadth-first search (BFS) strategies tend to explore the many domain assignments for a particular variable without preference, and all of these are considered before attempting to instantiate another variable. Depth-first search (DFS) approaches are more greedy, and attempt to find a complete assignment set as quickly as possibly, instantiating each variable once, iterating over domain values for a variable only when seeking an assignment consistent with those done before. Still other approaches have attempted to gain some of the benefit of each BFS and DFS by forcing BFS to "prefer" partial solutions closer to a complete solution. In this way the search progresses angle-wise through the search space with the left edge of an imaginary line showing the boundary of search tilted down to the left. In [11] such a strategy is described and is referred to as *Left Wedge* search (LWS).

In designing an interactive approach, we must be aware that since the analyst must identify and re-order or remove candidates that appear unpromising, a common reference frame is important for these candidates. Comparing candidates that have similar shared attributes is certainly more intuitive than comparing unrelated ones. For example, if an interactive approach were used with search, one would need to devise a strategy that carefully chose the order in which variables are instantiated prior to search. A changing of variable instantiation order throughout search would require the user to make judgements about groups of partial solutions that are "partial" in different ways. A constant, predetermined instantiation order of all variables, or of at least sets of variables would provide a better 'uniformity" over the elements of comparison.

Interactive BFS with a constant variable instantiation order would involve the analyst in removing candidate solutions early in the problem solving process, where few variables are yet instantiated. In addition, a BFS approach finds many partial solutions at a single "level" of detail first, saving them all before attempting to resolve ones at more detailed levels. This characteristic of BFS allows the analyst to remove candidates not to his/her liking as search progresses.

A LWS approach is simply "tilted" BFS, and as such retains the "storage" advantage of BFS, although elements are considered as a set not at a single level, but rather on the basis of some measure preferring more complete solutions at the expense of additional less complete ones. The tiltedness implies that elements under consideration at any given time will be of varying degrees of completeness, and so any interactive approach would require that the user make judgements between candidates of this type.

Depth-first search quickly locates more complete assignments, and provides the user with partial matches much closer to a complete solution more quickly, however, depth-first essentially thrusts towards a complete solution in a somewhat haphazard manner, and any interaction would require some stage or pause in search where the user may intervene and remove unlikely candidates or impose some order on the ongoing search.

We need a model that will provide us the ability to intermingle search, constraint propagation, and interaction in a flexible enough manner to accommodate changing or different users' needs. Abstraction offers this model, and in fact provides several possible solutions.

Abstraction and search

Search strategies can be either coherent across levels, considering all partial solutions at different abstraction levels according to the same measure, or they may consider partial solutions at different abstraction levels separately.

In a strategy such as LWS that was coherent across abstraction levels, it would be necessary to decide between locating additional partial solutions at a certain level, or pursuing some candidates at successive levels. This approach appears to be counter to the method of recognition employed by analysts. In a strategy that utilized different methods for search within an abstract level and search across levels, some set (possibly all) solutions at an abstract level would have to be found before solutions at subsequent levels are considered. In the case where all solutions at a particular level are found before advancing to the next level, if search were halted at an arbitrary point in time, all of the resulting "best-so-far" answers are on the same abstract level. However, time spent searching the less constrained CSPs might have resulted in more concrete solutions.

Search supported with non-hierarchical use of MOTORS is based on various depth-first strategies capable of finding one, some or all solutions. A unified hierarchical strategy might utilize a depth-first approach at each level, and a breadth-first approach across levels. The order in which abstract level solutions are considered between levels may be of importance to the analyst. For instance, if an analyst were searching a solution set for a particular solution, intervention in determining the order in which abstract solutions were examined could dramatically affect the search time required to find one particular solution or set of solutions.

Algorithm

Figure 6 details our interactive algorithm for pattern recognition based on abstraction. The following terms should be defined: Rep_i indicates the representation of the ith level of abstraction, where 1 is the most abstract, and N is the most specific; *Situation_Elements* indicates the set of individual sightings which are used to create the domain for each variable; $IS\_i \rightarrow j$ indicates the independent subproblems which result from the change of representation (Figure 4) between abstract level i and abstract level j; $CR\_i \rightarrow j$ indicates the constraint difference mapping between level i and level j; $CC\_i \rightarrow j$ indicates the new connective constraints which connect each IS_$i \rightarrow j$ to the level i partial solution.

Search commences with the most abstract representation, Rep_1. Initialization steps include initializing Domains for each Rep_1 variable according to each variable's characteristic constraints such as type or size, the representation change operations between each level (IS, CR, CC) are either identified or provided, and initial heuristic reduction of the Rep_1 problem by constraint propagation algorithms is performed. Search for all possible matches commences, using as input the ordered set of partial solutions from the previous abstract level (initially only Rep_1). A backtracking strategy is heuristically selected, and search commences. Use of heuristics during search within a level is discussed in detail for each option in [10]. During backtracking search, heuristics are utilized at Step 9 to select the next candidate node for further instantiation, at Step 11 to select a variable within a candidate node to instantiate, at Step 13 to determine how much (if any) constraint propagation is done to further simplify the remaining problem aspects based on the selection in 11, and at Step 14 to determine if any of the reductions performed at 13 should be propagated upward in the search space.

As the depth first strategy locates solutions to this abstract representation, these solutions are passed to the interactive *This_Level_Revise* process where the analyst can visualize and analyze the solutions on a workstation making use of other tools, and can then re-order or remove them according to his/her own criteria. Configurable critic heuristics can preview solutions, prior to handing them to the analyst, to filter out undesirable solutions according to analyst-indicated preferences. Duplicate sightings in close proximity can result in repeated solutions where the only difference is in a certain variable assignment. Further work is indicated in determining the spatial and other characteristics that indicate the "sameness" of these elements. However, this identification of nearby variable matches as part of a same-template instance can assist in the analyst's fusion task of these duplicate elements.

The analyst can allow the parallel search and review process to continue as long as desired, or until all abstract solutions are located. Once satisfied, the selected abstract solutions are passed to the next level for further refinement, and the process repeats for the next solutions. At the refinement stage, the representation change operators are applied, first pruning all partial solutions which fail to accommodate the constraint restriction between levels, second, initializing the independent subproblem variables according to variable characteristics, and third, extending the abstract solutions to less abstract problems via addition of the independent subproblems. The domains of the now attached independent subproblems can be reduced according to local heuristics, such as particular spatial constraints restricting the domain of the template based on the abstract solution in question, by selectively applying the connective constraints between abstract solution and independent subproblem, and by some heuristically selected constraint propagation method. Once simplified in this manner, all remaining problems are passed as input to a new interactive search phase. This process will continue until the analyst is satisfied with the results, or until all so-

```
Abstract_Interaction( Rep_1 ... Rep_N , Situation_Elements )

1 Create domains for Rep_1 variables, consistent with node constraints
2 For all level changes i->j
  3 Identify IS_i->j, Independent Subproblems added between level i and j
  4 Identify CR_i->j, Constraint Restrictions on level i constr at level j
  5 Identify CC_i->j, Connective  Constraints between level i and all IS_i->j

6 Reduce domains of Rep_1 variables via selected constraint propagation

// Search for solutions at level i, initially Rep_1
7 For each unsolved problem at level i, select backtracking strategy,heuristics

   8 Loop unless User_Interrupts
          Interrupt: save backtrack state position, exit Loop.
     9  Select backtracking node for expansion
     10 If Variables_Left to assign
        11 Select variable for instantiation
        12 Create successor from domain values for selected variable
        13 Propagate constraints using selected domain value
        14 Upward propagate if possible
        15 This_Level_Revise( solution )
     16 Loop (8) till done backtracking.

// Dynamic user interaction
This_Level_Revise( solutions )

  17 Delete solution if critic heuristic fails
  18 Delete solution if duplicate from noise
  19 Order  solution by ordering heuristic

  20 Analyst displays, removes solution(s) as required
  21 Re-order solutions according to analyst preference

  22 Repeat This_Level_Revise
        until User_Advance_Next_Level
     or until User_Select_Previous_Saved_Backtrack_Postition.

// Descend Abstraction Hierarchy i->j
  23 Applying CR_i->j, pruning all level i solutions that fail.
  24 Assign domains to IS_i->j variables, consistent with node constraints
  25 Extend all level i solutions to level j problems by adding IS_i->j

  26 For all j extended level i solutions
     // Local Consistency enforcement
     27 Simplify IS_i->j variable domains by applying some spatial constraints
     28 Simplify IS_i_>j variable domains by applying some CC_i->j
     29 Simplify IS_i->j variable domains by selected constraint propagation

  29 Repeat Search for each level j problem until j > N or USER_INTERRUPTS

  30 When j = N, remaining solution set is returned
```

Figure 6: Abstracting Interactive Search Algorithm.

lutions are found at the most specific representation level.

Final solutions can be utilized by the analyst and accepted into the situational view, thus relating identified template elements. Partial solutions can be instantiated also, with the elements not identified inserted into the situational view as expected but not sighted representations. The potential ranges of these unsighted elements can be restricted according to existing constraints with the identified elements, and according to other available information such as terrain, and analyst preference. The existence of these elements, while hypothetical, is supported by the analyst's acceptance of a suggested template. Further work in analyzing and dealing with analyst's expectations can be found in [9].

The role of abstraction in interaction

The human intuition behind using abstract problem representation and solution strategies is that in complex situations, it is often easier to attempt to "rough-out" solutions considering only the most important issues first, and then attempt to refine one of the "rough" solutions into a usable solution. We have attempted to design an interactive system of this type with a model of recognition familiar to the user.

Just as in human problem solving, a user can quickly glance over "high level" options and discard those which violate some constraint or condition not represented explicitly in a domain. Since any domain model is only approximate, we make use of abstraction guided interaction to assist the user in recognizing partial solutions that yet fail to resolve at a more detailed level as a result of an incorrectly or too strictly defined model.

We have attempted to create a model of interaction which demonstrates the following characteristics:

1. A user may be content at some point with some subset of all possible abstract solutions, may explore these at more detailed levels, and may wish later to attempt to locate additional abstract solutions.
2. The solutions at each level of abstraction should be found as quickly as possible in order to offer this level of solution as quickly as possible to the user as user consideration time may exceed computation time.
3. The user may re-order or remove solutions at each abstract level before search proceeds at a lower level. Solutions should be presented as discovered for consideration at a single level while others are still being found.
4. The user should be capable of interrupting or postponing the ongoing search at a single level and indicate "enough" solutions have already been found at this level, and search should then proceed on refining those found at the next more specific level. The interruption can indicate that the time available for searching has ended, and approximate answers are required immediately.
5. Search at each level should proceed in a depth-first but complete manner, finding this-level solutions as quickly as possible for consideration. Each member of the solution set at this-level is a candidate to be refined at the next level.

The development of an interactive system using abstract or hierarchical descriptions of the recognition process may be seen as an attempt to formalize part of expert template recognition. The use of hierarchies provides a method of expression which is capable of capturing domain information not represented in a non-hierarchical representation, and naturally allows for the integration of certain domain concerns such as signature equipment. The application of spatial heuristics for restricting the domains of partial solution problem extensions across hierarchical levels acts as a method of restricting possible specializations of abstract solutions in a fashion similar to the way an analyst limits his/her focus of attention.

Further investigation of the analyst's ordering and rejection of candidate solutions between abstraction levels may provide insight into future solution ordering or rejection heuristics. Analyst confidence in any suggestion oriented system is dependent on the ability of the system to justify suggestions in familiar terms. In the case where a partial solution is suggested in the absence of either affirming or denying indicators, the hierarchical resolution process can provide information about what specific information might be gathered in order to reduce the number of possible explanations of current sightings, or to increase the confidence level to the next defined hierarchical definition. Since the structure which determines the change of representation between an abstract level and the more specific is fully described, the process of explaining why a certain partial match qualified or failed to qualify at the next level is a simple matter of noting which additional constraint eliminated the solution.

Noise	Whole	Abstract
100	0.13	0.28
200	0.26	0.67
300	0.63	1.65
400	1.46	4.45
500	1.98	7.41
600	2.19	14.18
700	4.45	17.08
800	5.0	28.04
900	5.85	27.97
1000	9.6	45.51

Table 1: Maximum search time, two template models.

Experiments

Currently, all experiments have been undertaken with a limited implementation of the algorithm described which only functions at a single level of abstraction. However, this limited implementation representing steps 8 through 16 of our algorithm of Figure 6, has provided us the opportunity to explore several backtracking options, specific spatial heuristics, and varying degrees of constraint propagation before and during search. While detailed discussion of these results is left to a later publication and is described further in [10], Table 1 shows the CPU search time required to locate all occurrences of the template described in Figure 1, and an abstracted version of 4 Slots described in Section 3, in varying amounts of sighting noise. Each entry in Table 1 represents the worst CPU performance in 10 experiments. The "Whole" template had one occurrence placed in the data, while the "Abstracted" template had two occurrences placed in the data. Random noise resulted in none or 1 spurious solution for "Whole", and as many as 6 for the "Abstract".

We can conclude from this simple experiment that we must carefully determine the nature of our most abstract template so as to not overburden ourselves with solutions. Care must be taken that the elements of the most abstract template truly represent "critical" elements. Use of abstract templates which are under-constrained can result in far too many partial solutions at early search stages. One important aspect of our future work is to attempt to better characterize actual situational information to determine distribution and frequency of various sightings more accurately.

Conclusion

In this paper we have described a model for interactively recognizing spatial deployment templates in noisy and complex situational information. This model of interaction with the expert analyst is based on increasing confidence over time in the veracity of certain identified template instances. The ability of an implemented non-interactive system to provide single-level solutions in a reasonable time frame suggests that a strategy of most-abstract-solutions-first can be undertaken in a limited and predictable period of time.

Acknowledgements

Thanks to Dr. Qiang Yang of the University of Waterloo for providing comments on early aspects of this work, and for providing a skeleton lisp implementation of simple CSP search. Thanks also to Luc Lamontagne and Stephane Lapointe of DREV and to Kirsten Wehner for providing valuable comments.

References

[1] L. Des Groseilliers. Tactical information fusion prototype: Basic intelligence. Memorandum 3093/92, Defense Research Establishment Valcartier, Valcartier, Quebec, February 1992. Unclassified.

[2] E. Freuder and J. Wallace. Partial constraint satisfaction. *Artificial Intelligence*, 58:21–70, December 1992.

[3] Craig A. Knoblock. *Automatically Generating Abstractions for Problem Solving*. PhD thesis, School of Computer Science, Carnegie Mellon University, 1991.

[4] Vipin Kumar. Algorithms for constraint-satisfaction problems. *AI Magazine*, pages 32–44, Spring 1992.

[5] Steven Minton, Mark Johnston, Andrew Philips, and Philip Laird. Minimizing conflicts: a heuristic repair method for constraint satisfaction and scheduling problems. *Artificial Intelligence*, 58:161–205, 1992.

[6] Bernard Nadel. Representation selection for constraint satisfaction: a case study using n-queens. *IEEE Expert*, pages 16–23, June 1990.

[7] Bernard A. Nadel. Constraint satisfaction algorithms. *Computational Intelligence*, 5:188–224, 1989.

[8] NATO. Nato glossary of terms and definitions. Technical Report AAP-6(R), NATO, 1988. Unclassified.

[9] Graham Williams and Steven Woods. Representing expectations in spatial information systems: A case study. In *Proceedings of the 3rd International Conference on Large Spatial Databases*, June 1993.

[10] Steven Woods. Interactive recognition of spatially defined model deployment templates. Research memorandum, Combat Intelligence Automation Group, Defence Research Establishment Valcartier, Courcelette, Quebec CANADA, August 1993. unclassified.

[11] Steven G. Woods. An implementation and evaluation of a hierarchical non-linear planner. Technical report cs-91-17, Computer Science Department, University of Waterloo, 1991.

[12] Qiang Yang and Philip Fong. Solving partial constraint satisfaction problems using local search and abstraction. Technical Report cs-92-50, University of Waterloo, 1992.

[13] Qiang Yang, Josh Tenenberg, and Steven Woods. Abstraction in nonlinear planning. Technical Report cs-91-65, University of Waterloo, 1992.

404945

Toward Time-Dependent Planning of Missile Allocation and Engagement Scheduling in a Naval Threat Evaluation and Weapon Assignment (TEWA) System

Bruce A. Chalmers[1]
Situation Assessment and Resource Allocation Group
Command and Control Division
Defence Research Establishment Valcartier (DREV)
2459 Pie XI, Courcelette, PQ, Canada, G0A 1R0

#149748

Abstract

Planning the allocation of limited defence resources and their engagement scheduling is an important function of a weapon engagement manager in a supportive single-ship or cooperative multi-ship battle management system. The manager is an intelligent agent in a naval command and control (C2) system for threat evaluation and weapon assignment (TEWA) that must plan, coordinate and direct point or local area defence actions involving hardkill and softkill weapon systems to achieve mission goals. It has to operate in complex dynamic and uncertain battle environments under critical constraints on real-time performance. The system must be capable of interacting with the battle world and a commander (the supportee) in a responsive, reliable and predictable manner.

We present in this paper a model of a deliberative planning function embedded in a real-time layered control architecture for a weapon engagement manager in a naval anti-air warfare (AAW) TEWA system. We then specialize it to develop an adaptive utility-driven planning function for a surface-to-air missile (SAM) engagement manager for single or multiple naval platforms. Its deliberation, based on feedforward control, is concerned with conditionally planning and scheduling assignments to anti-ship missile (ASM) threats of SAMs and the use of illuminator resources to support the guidance of semi-active missiles. Because deliberation uses state information derived from predictions of the evolution of a dynamic world, plans are used to guide actions at lower levels of the control hierarchy, but not to control them. Computing an optimal plan is NP-hard, which makes it impossible to guarantee optimal response within real-time deadlines. To overcome this, we outline an adaptive approach based on off-line knowledge compilation, on-line classification of battle scenarios and almost anytime planning to permit a resource-limited agent to generate effective plans.

Introduction

Driven by the worldwide proliferation of ASMs and technological improvements in sensor, guidance and communication technologies, naval AAW defence systems in the nineties and beyond may be required to deal with high density threat engagements that occur in a timescale of minutes or seconds. This is leading to an enlargening of the size of the battle space as use of a variety of limited renewable and non-renewable defence resources is coordinated in real-time at both the single platform and cooperative multiplatform levels and allocated and scheduled to optimize protection of own ships and defended assets [1, 2].

Such resource allocation problems are an important component of naval battle management. Some problem formulations have previously been studied. For example, cooperative force level missile launch scheduling is conceptualized in [3, 4] as a deterministic production floor shop scheduling problem of minimizing the total weighted flow time, subject to time-window job availability and machine downtime unavailability side constraints. Formulations based on the use of Markov decision processes appear in [5-7]. Techniques for missile launch scheduling for single platforms are presented in [5, 6]. Selection of dynamically arriving tasks (threats) with deterministic deadlines and renewable resource allocation for a naval battle group are considered in [7].

Our research is addressing such resource allocation problems in the general context of developing a weapon engagement manager for a

---

[1] Phone: (418) 844-4691
E-mail: bruce@cc.drev.dnd.ca

supportive autonomous naval TEWA system that must plan, coordinate and direct in real-time point or local area AAW defence actions. Both single-ship and coordinated force level are being considered. Multiplatform resource coordination assumes a robust communications network and a computation capability that permit local information from a variety of distributed sources or sensors, including radar, electronic support measures (ESM) and identification of friend or foe (IFF) responses, to be continuously fused to provide a force tactical situation picture. Fully coordinated firing actions may then be achieved within a centralized or virtually centralized battle management control architecture. In favour of an evolutionary approach to the research, defence response is assumed for now to be layered according to the type of defensive systems employed in a layer and explicit consideration is given only to hardkill systems. Integration of softkill ship defence systems, including electronic countermeasures (ECM), decoys and chaff, is envisaged at a later stage.

Specifically, we present in this paper a model of a deliberative planning function embedded in a real-time layered control architecture for a weapon engagement manager in a naval AAW TEWA system. We then specialize this model to one for conditionally planning SAM firing and guidance actions against ASMs, categorized as sea-skimmers, shallow-divers, or high-divers depending on their attack profile. The reader may consult [3] for a more detailed description of the battle environment in which the manager may be expected to operate and of the opportunities for and the potential performance benefits to be derived by multiplatform coordination. The model is based on using temporal projection of the current tactical situation picture and a normative evaluation of potential histories of the battle world to formulate a conditional action plan. Optimal planning with respect to the model is NP-hard, which makes it impossible to guarantee optimal response within real-time deadlines. To address this, we also outline an adaptive approach based on off-line knowledge compilation, on-line classification of battle scenarios and almost anytime planning to permit a resource-limited agent to generate effective plans.

Block Specification and Layered Architecture

A weapon engagement manager is an intelligent agent in the TEWA system with a number of adaptive real-time performance requirements on its planning. For example, there is a threat-dependent hard deadline associated with processing each of a time-varying number of threats before it causes substantial damage. It has to dynamically interleave, and even overlap, incremental planning and execution, with (re)planning triggered by the occurrence of significant events in an internal model of a dynamic world in which new events that impact the plan horizon may occur at any time. It is a limited rational agent [8] since it is situated in a failure prone computing environment in which it competes for access to limited computational resources (processing time, space) for its real-time problem solving and deliberation. In addition, even if hard deadlines are respected, there is a soft real-time cost associated with the time spent on deliberation prior to acting. For example, the range of options for engaging a threat prior to its hard deadline (the battle space of the threat) generally diminishes with the passage of time. To illustrate this simply, consider that the number of feasible engagements in the period from birth to death of a hostile track is monotonically decreasing in the time of first engagement of the track. Finally, the quality and completeness of temporal projections (track trajectories, etc.) of the state of the battle world on which the manager's reasoning is based can vary from one moment to the next.

We are investigating an approach to designing a manager that satisfies these requirements. Two aspects of this approach are now briefly reviewed. The first, examined in the remainder of this section, is based on the use of a multi-layered real-time architecture. The second, examined in the next section, describes control parameters for engineering an adaptive real-time agent.

The principal idea behind the use of a layered architecture is to hierarchically decompose the functions of the manager depending on a number of considerations, including the level of abstraction in the internal world model used in a layer, the knowledge representation formalisms and type of reasoning involved, the nature of its control mechanisms, the temporal characteristics and real-time constraints on the inter- and intra-layer time-dependent information flows (fluents), as well as the transformations on these fluents. A fundamental consideration is the principle of Saridis [9] of combining "increasing intelligence with decreasing precision", whereby a plan generated at one level of the hierarchy is used to influence actions at a lower level that interacts more directly with the dynamic battle world, but not to control them. There is an extensive literature within the areas of artificial intelligence and intelligent control dealing with layered architectures as a basis for designing intelligent agents for control systems.

Fig. 1 presents a simplified block specification of a two-layered real-time control architecture for the manager. It concentrates on showing the decomposition of the functions (blocks) and information flows (arrows) mentioned in this paper. Not shown, for example, are fluents associated with potential interactions between the manager and the supportee via a man-machine interface (MMI) manager in the C2 system. Such interactions would be based on a definition of the division of responsibility, depending on operational context, between the system and the supportee.

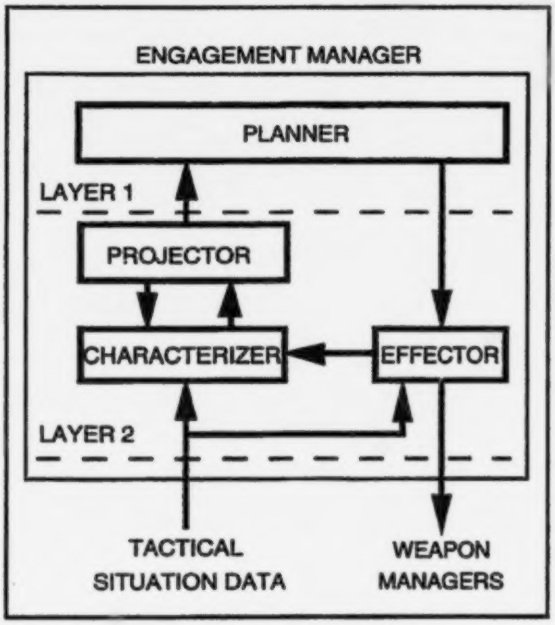


FIGURE 1: Layered architecture.

The *planner* computes an effective plan for assigning and scheduling weapons and guidance systems over some time horizon, subject to firing doctrine and resource availability. It provides service in response to the occurrence of an event arising from a significant change in the tactical picture that requires deliberative (re)planning. Recognizing such significant events within temporal constraints is handled by the *characterizer*, using the internal picture of the battle world and information from the projector and effector. The *projector* extrapolates threat tracks over a forecast horizon. It uses this information to predict when potentially hostile tracks are engageable, from which platforms and by which weapons on such platforms, as well as measures of effectiveness of such strike actions. By communicating with the weapon managers, the *effector* coordinates and directs the execution of plans received from the planner and monitors outcomes of strike actions.

The planner bases its plans on projections from the projector. Such projections will necessarily be defeasible due to statistical error and incomplete or imprecise information, leading to nonmonotonic reasoning. However, they are assumed to be perfect at the level of world abstraction of the planner. An implication is that the effector must implement plan repair strategies when real-time deviations from projections are "small". Hostile tracks that cannot be engaged in this manner must be fed back via the characterizer for deliberative replanning; an example would be a situation in which the effector drops a hostile track from its list of active engagements because it is unable to satisfactorily resolve a real-time resource contention that was not present in the projections on which its higher-level action recommendations were based.

Tactical situation fluents are related to a number of assessments of the threat, the defence force and its mission [10]. They also provide control parameters needed for the next planning episode. There are two data paths in the architecture of Fig. 1 leading from the arrival of such fluents to the weapon managers. The first is via the effector in layer 2 only. This path implements *feedback control*. Fluent updates along this path are periodic (individual fluents may be of different periods), driven by sampling and processing of dynamic real-time sensor information. Reactive responses to changes in the tactical picture are effected. The second data path is via both layers 2 and 1 and implements *feedforward control*. Fluent updates from the projector via the planner to the effector are asynchronous and fluent transformations are of a deliberative nature. While response to changes in the tactical situation along this path is expected to be fast, it is not on the same timescale as that on the other.

Real-Time Specification

We are concerned here only with the deliberative planning function in a manager. A traditional approach to real-time system design is to "guarantee a response after a fixed time has elapsed, where the fixed time is provided as part of the problem statement" [11]. While this approach may be appropriate for real-time specification in layer 2 of our architecture, it is inappropriate in layer 1 in view of its adaptive real-time performance requirements. Moreover, as Zilberstein [12] points out, "this conservative approach leads to inflexible systems that may be under-utilized since in many domains there are no clear, rigid deadlines". He proposes an alternative view of real-time systems based on operational rationality. Its thesis is that "the fundamental property of an operationally rational agent is the capability to vary its deliberation time according to time pressure".

Our real-time specification of the planner is motivated by the above considerations. It provides specific control parameters for engineering an adaptive real-time manager. In addition, further incorporation of Zilberstein's notions of a global time-dependent utility function of an agent (to dynamically measure the performance benefit and cost of planning) and of its conditional performance profile (as a function of the quality of its input fluents) suggests an interesting extension of the specification aimed at developing a metaplanning framework for the manager. This framework would explicitly account for the effect on the quality of the manager's performance of the quality and timeliness of the tactical situation picture it receives; also, it would permit the metaplanning required to actually tradeoff in real-time plan quality, computing time devoted to planning and the allocation of computational resources. Developing and embedding such a framework in the architecture is a subject of ongoing research.

Timing facilities for maintaining wall-clock time to permit synchronization with events in the battle world are assumed to be supported. On a service call, the deliberative planner receives a temporal projection of the force tactical situation picture for a forecast horizon up to time T. This projection is derived from the internal perception of the history of the world up to some elapsed time t. The planner may provide service in one of two anytime modes. In contract mode, its service is to produce in Δt units of computing time an effective plan for a plan horizon within the forecast horizon; Δt is specified at the time of the service call. The plan will be eligible for use by the effector commencing at time $t+\Delta s$ (the start of the planning window for the service call), where $\Delta t < \Delta s$, and it must be consistent with defence commitments up to this time. In interruptible mode, Δt is not specified to the function which may be interrupted unexpectedly. The interruptible mode has an advantage over the contract mode of providing for more run-time scheduling flexibility and it can cope with time pressure created by unpredictable variability or degradation in computational resources.

Values of Δt, Δs, T and the plan horizon are determined dynamically and can be used to engineer an adaptive real-time agent. For example, Δt can depend on time-varying conditions in the battle world, the current processing and communication load of the system, and system timing requirements that may be imposed to allow, depending on battle context, interaction and positive control by the supportee prior to plan implementation. Δs can control the sensitivity of the agent's response to unexpected run-time events. T can be based on estimates of the current quality of projections of the evolution of the battle world (quality may be measured by uncertainty ellipsoids, for example), accounting for loss in accuracy of predictions as they extrapolate into the future. In addition, T can be influenced by an estimate of the completeness of projections (based on run-time monitoring) and can be used to bound plan depth when the velocity of unanticipated change in the world is so high or its behavioural inertia so small that deep planning is ineffective and a waste of computational effort. The length of the plan horizon depends on Δt and on the computational performance profile of the planner.

Conditional Planning Model

The SAM engagement model assumes one or more ships under attack by ASMs. Support ships, of which there is at least one, can perform the functions of surveillance/tracking, SAM launching for point or local area defence in the missile engagement layer, point defence in their close-in layer, kill assessment, and radar support (when necessary) for missile guidance. Included are home-all-the-way (HAW) guidance and mid-course guidance with semi-active terminal homing. Missile firing actions can be shoot-look-shoot or some related generalization of this doctrine involving salvos (e.g., shoot-shoot-look-shoot); firing policy may even use variable salvo size, dynamically determined from one planning episode to the next.

We now present an unbounded rational agent model for the planning function of a SAM engagement manager. Its application assumes a dynamical system representation of kinematic entities in the battle world that over the forecast horizon of a planning episode is reasonably inert. This is to permit its dynamic process model to be estimated with adequate accuracy to justify deliberative reasoning about actions over the horizon. Firing actions are capable of influencing this inertia, subject to uncertainty on the effects of such actions. Plans are conditional on outcome assessments available to the effector at plan execution time. This is aimed at reducing the frequency of deliberative replanning arising from unsuccessful engagements. The price of this approach, however, is potentially greater computational load in generating conditional plans. (An alternative approach, based on unconditional planning, is also currently being examined for purposes of comparison.)

Interaction by the manager with the battle world is modelled as the evolution of a history H [13] over time. H is a mapping from timepoints to the internal representation of the state space of the world. The portion of the history that has been realized up to the current time t is obtained from the perception of the world state contained in the tactical situation picture. The conditional planner is assumed to provide control plans related specifically to missile firing and guidance actions over some plan horizon. A defence plan P, derived from H, is a conditional plan represented by a labelled directed tree. A node with one direct successor corresponds to a defence action/activity. A defence activity takes place over some time interval. An example is continuous wave or interrupted continuous wave illumination of a specific threat by a multifunction radar/illuminator (MFR/I) to enable guidance of a semi-active missile so that it homes to its intercept with the threat. An activity is represented by two discrete defence actions corresponding to starting and ending the activity. A defence action includes these actions as well as missile firing actions. Each firing action involves a specific ship-threat pairing. A node with two or more direct successors, depending on the granularity of the stochastic threat destruction or ship damage model used, is a conditional node corresponding to a stochastic outcome of either a firing action (consequence-of-action uncertainty at action time) or a threat strike on an own ship, including both support and protected ships. Nodes are labelled with their time of occurrence and the state of the battle world at that time. For a plan to be feasible, the time-stamped action/outcome sequence implied by successive nodes on any directed path in the plan tree must be realizable; in particular, this means that timing behaviour determined by kinematic laws governing missile and threat motion must be respected and firing doctrine and renewable and non-renewable defence resource constraints on fire control radars/illuminators and SAMs must not be violated.

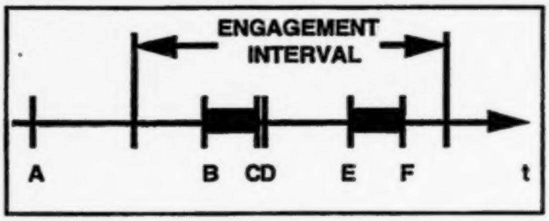


FIGURE 2: Action time line.

Fig. 2 illustrates an action time line of a conditional plan for an engagement involving a single support ship with (at least) two SAMs and a single threat. Illumination is assumed to be feasible in the time intervals indicated (shown shaded). The SAM system uses mid-course guidance and semi-active homing. A SAM is fired at A. It receives command guidance from the ship in A-B, is illuminated in the terminal phase B-C for a possible intercept at C. C corresponds to a conditional node in the plan tree, representing the outcome of the firing action at A. If intercept fails, a second SAM is fired at D, receiving command guidance in D-E and illumination in E-F. The SAM engagement interval, defined as the interval from first feasible time of intercept to last feasible time of intercept, is also indicated. A-C corresponds to the fly out time of the first SAM and D-F to that of the second when the second firing action is initiated.

Based on resource availability and control on a planning episode, the goal of the unbounded rational agent is to choose a most preferred feasible plan over the plan horizon. In a decision-theoretic framework, given the probability distribution on the outcome states, $s \in S$, of the world at the end of the plan horizon and a utility function u on outcome states, the agent chooses an optimal plan, P^*, according to:

$$P^* = argmax_P v(P), \tag{1}$$

where the value $v(P)$ of plan P is the expected projected value of u resulting from its execution, given by:

$$v(P) = \sum_{s \in S} prob(s \mid P) u(s). \tag{2}$$

The utility function u may depend on the planning episode. It captures the supportee's current preferences among a variety of individual attributes of world state, depending on mission goals, as well as his disposition or aversion to risk. Mission goals may include: maximize threat value destroyed (subtractive defence), maximize value of surviving assets (preferential defence), and minimize wastage of missiles. The setting of a vector-valued preference measure on world states may be obtained by generalization.

Starting from a perception of the tactical situation at the current time, there may be infinitely many histories that describe all potential evolutions of the battle world thereafter, each consistent with the current perception and each determining a feasible conditional defence plan. This comprises the set of all potential histories [13]. For example, in Fig. 2, firing actions may be rescheduled in any way that satisfies firing doctrine, illuminator availability and the requirement that potential intercepts occur in the engagement interval. This suggests that the optimization in (1) is, in general, with respect to an infinite set of feasible plans. Obtaining a combinatorial optimization problem over a finite plan space requires narrowing attention to a finite subspace. This procedure, which produces a potentially suboptimal plan, currently proceeds in two steps:

Step I

Generate a finite set A of defence actions, each labelled with a time of execution during the current plan horizon.

Step II

Determine a best plan P^* from the restricted finite set $\Pi(A)$ of feasible plans whose actions occur in A; that is,

$$P^* = arg\,max_{P \in \Pi(A)} v(P). \tag{3}$$

Two approaches to achieving Step I are to exploit some characterization of defence actions in an optimal plan or to use a heuristic selection procedure. The advantage of the former is that it leads to an optimal plan. The second may be used to reduce computational complexity. A hybrid approach is also possible. An example of the first approach is given in the next section. An example of the second would be to choose actions from several time periods within the plan horizon to achieve defence-in-depth.

Step II rests on an encoding of all members of $\Pi(A)$ as solution trees of a finite AND/OR decision tree T. In addition, P^*, defined by (3), corresponds to an optimal solution tree [14]. T can be explicitly generated as needed via temporal projection. This procedure can be viewed as a form of simulation-based planning [15] in which a search is made through a variety of simulations of plans to select the plan with the most successful simulation. The nodes of T are labelled with a time-stamp and state information on the battle world at that time. Various data structures are also maintained at each generated node to permit enforcement of consumable and contentious resource constraints and firing doctrine in simulations of all histories that emanate from a node. Interior nodes of T are of two types, decision or outcome. A decision node is an OR-node. It corresponds to a specific action in A and the two edges emanating from it represent the choice to perform the action or not. An outcome node is an AND-node. It corresponds to the outcome of a specific previous firing action or of a specific threat strike on an own ship. Such a node has two or more direct successors, depending on the granularity of the stochastic threat destruction and ship damage models used. Emanating edges partition the finite set of potential histories according to the outcome, which is unknown at the time of planning, and they are labelled with outcome probability measures supplied by the damage model. The value $v(P)$ of plan P is recursively computed (a foldback analysis) as a composition of the values of its subtrees, using the values of u on the world states at leaf nodes.

Various combinatorial search algorithms for Step II are being designed and tested. They are currently based on top-down or bottom-up search methods [14, 16, 17]. To reduce the run-times of these algorithms and to permit deeper search in the plan space for a given amount of computing time, versions for implementation on single-instruction multiple-data (SIMD) and multiple-instruction multiple-data (MIMD) parallel computer architectures are also being tested.

In general, determining an optimal conditional plan for the model is an NP-hard problem. An illustration in a naive deterministic setting is obtained as follows. There is one support ship with one fire-control radar to provide HAW guidance. SAMs are perfectly reliable (kill probability of unity) and u is given by the number of engaged threats. Each threat or missile moves at some constant speed along a straight-line trajectory. Then the decision problem of determining whether the fire-control radar can be scheduled to process (engage) all tasks (threats) is NP-complete, even if all tasks have the same release time and two distinct deadlines [18].

An Example

The example of this section leads to a multiplatform generalization of the single platform technique in [5], while also accounting for threat hits on an own ship.

A number of assumptions are made. They include: deployment against the same threat or different threats are independent; fly out time is a monotonically decreasing function of time of intercept; kill probability is a monotonically increasing function of time of intercept; the utility function u is a weighted sum of the number of threats eliminated before the end of the plan horizon - a threat weight reflects the value assigned to its elimination. In addition, we assume that the critical renewable defence resource constraint is due to the contention arising from a limit on the number of time intervals of guidance (illumination) activity from each support ship that can overlap at any time. Illumination from a support ship may be provided by multiple mechanically steered or dedicated electronically steered fire-control radars or by splitting or time-sharing the energy from a single MFR/I according to an energy budget. For HAW guidance, the resource is used (nonpreemptively) just prior to the time of fire until the end of intercept; for a mid-course guided missile, it is used during the terminal phase only.

It can be shown that under these assumptions defence actions in an optimal plan can be chosen from a finite (but potentially large) set characterized by an extremal property. Briefly, this property says that firing actions against each threat should be as late as possible, subject to firing doctrine and resource availability. For example, the plan in Figure 2 is dominated by the one obtained by shifting the illumination intervals and associated firing actions as far to the right as possible. Sequential and parallel algorithms based on enumerative backtrack search and temporal line sweep (backwards in time) are currently being used to identify elements of A for this example.

Toward Time-Dependent Planning

The planning problem and the real-time specification for its solution have been developed in a framework in which the manager is a time-dependent planner [19, 12], capable of varying its deliberation according to time pressure. But the specific planning model presented in this paper would require for its implementation an unbounded rational agent that has the luxury of ignoring the cost of deliberation (cost of time consumed and computational resources committed during deliberation). How can this immense gap be bridged? This section outlines one approach to resolving this problem that is being considered. An interesting feature of the approach is that it is model and machine independent. It may be applicable regardless of the specific algorithmic implementation of the planning function that is adopted in the real-time architecture for the manager.

The underlying idea is to control the complexity of plan generation by using a rolling plan horizon whose size is determined at the time of a service call. Depending on the anytime mode of the planner (contract or interruptible) and on its implementation, this may require computing a plan for each of a deepening sequence of plan horizons within the forecast horizon. The motivation depends on an assumption (which remains to be tested in experiments) that an effective plan of immediate actions should be based on looking ahead over a time horizon that is as large as possible. But the computational complexity of look-ahead is a function of the size of the plan window. This suggests a tradeoff. The approach is illustrated in Fig. 3, where l denotes the size of the plan window. Note that we are not concerned here with the choice of the forecast horizon, this choice having already been made prior to invoking the planning function based on considerations indicated in the real-time specification.

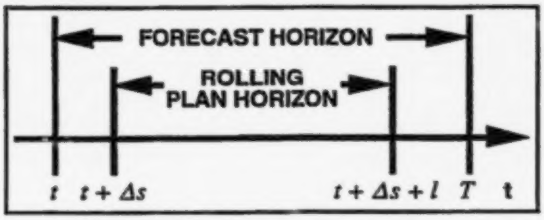


FIGURE 3: Rolling plan horizon.

The approach is reminiscent of one employed in real-time single-agent heuristic search [20] for obtaining a commitment to search moves in constant time. However, there are a number of important differences. Unlike heuristic search where the search process can be naturally limited to a discrete number of levels of look-ahead, our setting requires varying the plan horizon over a time continuum. Also, for a fixed length of plan window, the problem is further complicated by the fact that a varying number of levels of the search tree can fall in the window, depending on the particular battle scenario being processed. Finally, fixing the battle scenario and varying the size of the plan window can lead to disjoint portions of the space of feasible plans being searched.

In the following, M denotes a specific implementation of the planning function. We denote by $\rho(M,c,l)$ the computing time required by M to produce a plan in response to the scenario c (supplied by the projector) over a window of size l. For the kind of implementation we have in mind, it is reasonable to assume that ρ is monotonically increasing in l.

We consider first the case of the planner in contract mode, limited to Δt units of computing time. The planner will maximize l subject to $\rho(M,c,l) \le \Delta t$. It can quickly "solve" this problem with a table look-up if it has precompiled a performance profile table for a variety of sizes of plan window. Since it is clearly impossible to do so for every choice of c, an alternative statistical approach is reasonable. Battle scenarios have a number of natural attributes upon which a partitioning of the input problem space with respect to ρ may be based. Examples include various ASM and SAM parameters, attack profile, force and protected unit parameters, etc. An off-line Bayesian-based statistical technique for attempting this partitioning (class identification and description) is suggested by work in [21]. However, run-time monitoring of the battle world is now needed to make an on-line identification of the specific class C of scenarios confronting the planner when presented with a particular scenario c so that $c \in C$. We can then reformulate the tradeoff analysis of the planner as follows. It introduces the idea of an *almost anytime contract planner* in the sense that its anytime property is modulo a certain level of confidence. Let $0 < r < 1$ be an acceptable level of confidence that the planner will deliver a plan in Δt units of computing time. Then it will maximize l subject to $prob(\rho(M,c,l) \le \Delta t \mid c \in C) \ge r$. As before, this problem can be "solved" with a table look-up if it has precompiled probabilistic performance profile tables for a variety of sizes of plan window and for each scenario class. This is an off-line knowledge compilation process. Neural net technology functioning essentially as an associative memory may be a natural candidate for automating both the on-line scenario identification and table look-up process. An extension of the approach that increases the expected size of the plan window returned is to schedule a number of concurrent instantiations of an almost anytime planner, with confidence levels chosen from a finite strictly decreasing sequence.

Finally, Zilberstein's technique [12] of repeatedly invoking a contract algorithm with exponentially increasing time limits to obtain an interruptible algorithm is readily adapted to our probabilistic setting to produce an *almost anytime interruptible planner* with confidence level r from an almost anytime contract planner that has an equal level of confidence. The overhead of this technique is reflected in a loss of expected size of plan window for a given amount of computing time.

Summary and Ongoing Research

We presented in this paper an agent model for a deliberative planner in a real-time engagement manager. Further details appear in [22]. The model is being used to provide a basis for evaluating several performance aspects, including effectiveness of defence response and the impact of defence and computational resource limitations and of intelligent adaptive control, as well as to assess improvements from using a variety of parallel architectures to implement the manager.

References

1. Naval Forces Update, "Co-operative engagement and self defence," *Jane's Defence Weekly*, pp. 32-34, 19 June, 1993.
2. C. Beal, "Anti-ship missile technology: leaving well enough alone?," *International Defense Review* 10, pp. 957-964, 1992.
3. D. B. Boyer, E. L. Perry, E. L. Price, and S. Godrey, "Force level engagement control in naval anti-air warfare," in *Proceedings of the 1990 Command and Control Research Symposium*, Monterey, California, pp. 204-211, 1990.
4. H. D. Sherali, Y. Lee, and D. D. Boyer, "Models and algorithms for scheduling target illuminators in naval battle group anti-air warfare scenarios," to appear in *Naval Research Logistics Quarterly*, 1993.
5. M. D. Diamond and O. M. Carducci, "Decision processes for large scale resource allocation problems," in *Proceedings of the 8th MIT/ONR Workshop on C^3 Systems*, Cambridge, Massachusetts, pp. 153-160, 1984.
6. M. Diamond, C. Rennolet, J. Kimbel, and M. McNatt, "Optimal resource allocation and optimal launch scheduling for anti-air weapons systems," in *Proceedings of the JDL 1990 Symposium on C^2 Research*, Monterey, California, pp. 194-203, 1990.
7. P. B. Luh, Xi-Yi Miao, S-C. Chang, and D. A. Castanon, "Stochastic task selection and renewable resource allocation," in *Proceedings of the 9th MIT/ONR Workshop on C^3 Systems*, Monterey, California, pp. 59-64, 1986.
8. S. Russell and E. Wefald, "Principles of metareasoning," *Artificial Intelligence*, 49, pp. 361-395, 1991.
9. G. N. Saridis, "Toward the realization of intelligent controls," *Proceedings of the IEEE*, 67(8), pp. 1115-1133, 1979.
10. J. A. H. Miles, E. M. England, H. C. Faulkner, and S. P. Frampton, "Knowledge-based techniques for tactical situation assessment," in *Application of Artificial Intelligence to Command and Control Systems*, ed. C. J. Harris, Peter Peregrinus Ltd., pp. 315-355, 1988.
11. T. J. Laffey, P. A. Cox, J. L. Schmidt, S. M. Kao, and J. Y. Read, "Real-time knowledge-based systems," *AI Magazine* 9(1), pp. 27-45, Spring 1988.
12. S. Zilberstein, "Operational rationality through compilation of anytime algorithms," *Report No. UCB/CSD 93/743*, University of California, Berkeley, California 94720, 1993.
13. Y. Shoham, *Reasoning about Change, Time and Causation from the Standpoint of Artificial Intelligence*, The MIT Press, Cambridge, Massachusetts, 1988.
14. V. Kumar, D. S. Nau, and L. N. Kanal, "A general branch-and-bound formulation for And/Or graph and game tree search," in *Search in Artificial Intelligence*, ed. Kanal and Kumar, Springer-Verlag, pp. 91-130, 1988.
15. D. P. Miller, "A task and resource scheduling system for automated planning," *Annals of Operations Research*, 12, pp. 169-198, 1988.
16. T. Ibaraki, "Generalization of alpha-beta and SSS* search procedures, *Artificial Intelligence*, 29, pp. 73-117.
17. V. Kumar, "A general bottom-up procedure for searching And/Or graphs," in *Proceedings of the National Conference on Artificial Intelligence (AAAI-84)*, Austin, Texas, pp. 182-187, 1984.
18. K. I-J. Ho, J. Y-T. Leung, and W-D. Wei, "Complexity of scheduling tasks with time-dependent execution times," to appear in *Operations Research Letters*, 1993.
19. T. Dean and M. Boddy, "An analysis of time-dependent planning," in *Proceedings of the Seventh National Conference on Artificial Intelligence* Minneapolis, Minnesota, pp. 49-54, 1988.
20. R. E. Korf, "Real-time heuristic search," *Artificial Intelligence*, 42, pp. 189-211, 1990.
21. P. Cheeseman, J. Kelly, M. Self, J. Stutz, W. Taylor, and D. Freeman, "AutoClass: A Bayesian classification system," in *Proceedings of the Fifth International Conference on Machine Learning*, pp. 54-64, 1988.
22. B. A. Chalmers, "An adaptive planner in a layered real-time architecture for an engagement manager in a naval threat evaluation and weapon assignment system: An overview," DREV document, 1993.

90C 20C

Artificial Intelligence Applied to Sonar Data Management[1]

Martin Laplante,[2] Ph.D.
RES Policy Research Inc,
Suite 501, 100 Sparks Street, Ottawa

Gail E. Harris,[2] M.Sc.
RES Policy Research Inc,
Suite 501, 100 Sparks Street, Ottawa

#149749

Abstract

A recently completed study of how artificial intelligence techniques could be applied to the sonar data track association problem proposes a blackboard system of cooperating knowledge sources to assist an acoustic operator in analysing signals. The study analyzed the association of acoustic data (both narrow band and broad band) from a towed array or from a group of sonobuoys, and addresses both single and multiple sensor associations within the context of the Generic Sonar Architecture. The blackboard stores representations of segment associations along with their hypothesized dynamic model. The knowledge sources identified would use a variety of techniques including fuzzy logic, temporal logic, rule based reasoning, unsupervised neural networks, time delay neural networks, frame based reasoning and case-based reasoning. The overall blackboard architecture and the individual knowledge sources are described.

Introduction

Currently, analysis of sonar information is carried out by experienced acoustic operators who rely on accumulated knowledge to determine what signals are significant and what, if any, associations can be created between them. The operator can learn more about the target or targets generating the data by associating the observed signals together. Now the operator is better able to employ more advanced analysis tools.

As targets become quieter and more sophisticated, and both sensor and signal processing technology advance, the task of operator data association becomes more and more difficult. This is due primarily to the increase in volume and complexity of available data. To further increase performance and capability, future sonar systems must incorporate processes that assist the operator in data association.

The Generic Sonar Architecture was conceived as a platform for developing sonar analysis systems primarily for Anti Submarine Warfare (ASW) applications. The generic architecture consists of four major subsystems, as shown in Fig. 1. Most current research focuses on the Acoustic Data Interface (ADI) and Signal Processor (SP), however this paper concentrates on the right half, the Data Manager (DM) and the Display Processor (DP).

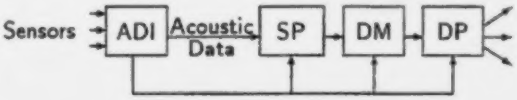


Figure 1: Generic Sonar Architecture

Acoustic data, received from sensors such as sonobuoys and towed arrays, is displayed in various formats, primarily as *narrow band* (NB) and *broad band* (BB) data. **NB** is a mapping of time versus frequency in the narrowband; bearing information also may be available but not displayed. **BB** is a mapping of time versus bearing.

The DP allows a *signal follower* to be assigned to a strong signal. The signal follower will create a *segment* which can be described by the time interval and frequency and/or bearing of the signal for each time slice within the interval.

Background

The study [1] was sponsored by the Canadian Centre for Marine Communications (CCMC). They brought together the prime contractor, Iotek, and the study subcontractor, Rockwood. CCMC envisioned Iotek's knowledge of sonar being complemented with Rockwood's knowledge of Artificial Intelligence (AI). Iotek in turn brought CCMC and Rockwood to Defence Research Establishment Atlantic (DREA) who has scientific expertise in both areas and who represented an end user. DREA agreed to participate in a study that would examine the application of AI techniques to the sonar data management problem.

---

[1] This study was conducted as a subcontract to Iotek Incorporated, 1127 Barrington St, Halifax

[2] formerly of Rockwood Informatics

Problem Definition

Given two segments are they associated? Associated usually means the segments showing on a display are signals from the same target. In the case of multiple ships moving as a group the group as a whole is treated as one target.

To perform segment association the operator will identify significant signals on NB and BB displays, and assign signal followers to them. The signal followers will provide the segment data to the association process. The association process looks at pairs of segments and assigns probabilities of these pairs being associated, based on the data that makes up the segment, available non-acoustic data, and appropriate specialized knowledge. We assume but do not rely on external target classification and localization processes to help rule out potential associations.

The segment association problem naturally splits up into seven sub-problems, which are discussed in the next sections.

- Temporal Association
- Intra-sensor Association
  - BB to BB
  - NB to NB
  - BB to NB
- Inter-sensor Association
  - BB to BB
  - NB to NB
  - BB to NB

Temporal Association

Given two segments from the same display determine if they are from the same target but received at different times (see Fig. 2). The process for solving this data association problem may require the raw input used to create the display, particularly for NB displays from sonobuoys where bearing information is available but not displayed.

Two segments separated by a significant time gap can occur for many reasons including:

- an acoustic source is at the edge of the range of a sonobuoy,
- a towed array passes through different convergence zones or receives signals that have bounced off the sea bottom

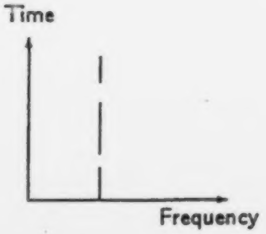


Figure 2: Multiple NB segments, separated by time gaps.

- a piece of machinery has been turned off and then on again,
- background noise masks the signal for awhile,
- a second acoustic source has come into range, or
- the signal was lost while the source or own-ship was manoeuvreing

In most cases where the signal is coming from a single target, the two segments will form a line at the same frequency or within a reasonable doppler shift. For narrow band displays from a sonobuoy this line will likely be straight, unless the segments are separated by the *closest point of approach* (CPA). For a broad band display from a sonobuoy this line should be straight along some slope for a ship travelling on a consistent heading, except when the segments are separated during the time the ship passes very near or directly over the sensor.

Intra-sensor Association

Intra-sensor associations are made when there are multiple segments from the displays for a single sensor. These arise for many reasons including:

- harmonic oscillation,
- multiple propagation paths,
- multiple noise sources from a single target, and
- multiple noisy targets.

Each of the sub-problems below also has a temporal variant, where the first segment ends before the second segment begins.

Broad Band to Broad Band

Determine whether two BB segments from the same sensor (sonobuoy or towed array) are associated. For the towed array the segments may have arrived along different propagation paths therefore these associations require knowledge about propagation paths. For the sonobuoy case it is most likely that there are two targets.

Suppose a single ship is in range of a towed array. It is possible the broad band display indicates two quite different bearings if the signal is arriving along multiple propagation paths such as bouncing off the sea bottom or refracting to create a convergence zone. If multiple propagation paths can be ruled out then it is likely that there are multiple targets being detected.

Narrow Band to Narrow Band

Determine whether two NB segments from the same sensor (sonobuoy or towed array) are associated. For the towed array case the segments may be on the same or different beams and for the sonobuoy case there is bearing information that is not displayed. For a towed array, these kinds of associations use knowledge about harmonics, similar patterns, maximum value for the rate of change of bearing, heuristics about multiple propagation paths and hints from classification. For a sonobuoy knowledge of harmonics is used, sometimes heuristics about multiple propagation paths and hints from classification.

If one ship is being detected by a towed array it is possible to have multiple segments at different frequencies and along different beams. The two segments could be from different machinery on one ship or the segments could be harmonic oscillations in which case frequency changes will appear as similar patterns. Two segments could also occur if the signal has bounced off the sea bottom as well as refracted to create a convergence zone (thus the sensor has received two signals possibly by two different beams).

Two NB segments could occur if two ships are being detected by a single sonobuoy or if the ship has a variety of noise sources.

Broad Band to Narrow Band

Determine whether two segments, one from a BB display and the other from NB, are associated. The reasoning is similar for both sonobuoys and towed arrays, both use knowledge about the geometry of the situation. For example, if the bearing information from the NB data is the same as that from the BB then strengthen the association.

Suppose a single ship is present and it has a variety of machinery making noise. Thus a NB segment could occur from say the engine and that combined with the rest of the noise causes a BB signal. If both signal sources come from the same general direction then it is likely that only one target is present.

If there are several ships travelling as a group then a BB and NB segment could arise. Until there is evidence to suggest otherwise this group will be treated as a single target.

Inter-sensor Associations

Inter-sensor associations need to be made when dealing with multiple sensors, as in the sonobuoy case. We are assuming that sonobuoys receive both broad band and narrow band signals. We restrict the analysis to a multiple target scenario with multiple sonobuoy sensors only.

Broad Band to Broad Band

Determine whether two segments on the broad band displays from two different buoys are associated. These kinds of associations use knowledge about the relative positions of the two buoys, the dynamics of ship motion and hints from the results of a localization algorithm.

If one ship is moving through a sonobuoy field and is within range of two different sonobuoys then we may have two segments as shown in Fig. 3. The bearing values can be quite different, depending on the relative positions of the sensors and the target ship. The magnitudes of the rates of change of bearing can be similar, within some tolerance, if the ranges from the sensors to the ship are similar.

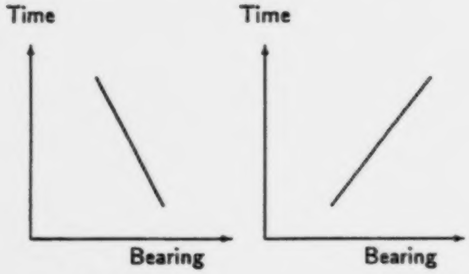


Figure 3: Sample of BB displays for two different sonobuoys.

If two ships are moving through a sonobuoy field then we may have again two BB segments from two sonobuoys. The association process must differentiate between this case and the above case using knowledge about geometry, ship motion and hints from a localization algorithm.

Narrow Band to Narrow Band

Determine whether two segments showing two frequencies from two different buoys are associated. These kinds of associations use knowledge about how the frequencies could shift, knowledge about the maximum difference between frequencies of segments from the same target (based on maximum ship speed), knowledge about the relative positions of the two buoys, knowledge of the dynamics of ship motion and hints from the results of a localization algorithm and/or a classification system to determine if the signals are from the same or different targets.

Segments from NB displays can show visual patterns, such as frequency shifts from machinery or knee bends at the CPA. These patterns may exhibit similar properties on different buoys thus knowledge of pattern similarities may be used.

If one ship is moving through a sonobuoy field and is within range of two different sonobuoys then we may have two segments. The frequencies could be similar if the ship is moving towards both buoys. If the ship is moving toward one buoy but away from the other buoy then the frequencies are likely to be different because of a doppler shift, but the total difference will not exceed some maximum and the rates of change of the respective frequencies should be similar (since the range of a sonobuoy is limited)..

Broad Band to Narrow Band

Determine whether two segments, one showing broad band information and the other showing narrow band information, are associated. The frequency information is generally ignored in this situation. The association problem then becomes the same as for broad band to broad band.

Approach

The problem of associating pairs of segments (and later groups) can be divided into several interacting subproblems, each of which can be solved using AI techniques. The subproblem knowledge systems can use, in their reasoning, associations from other subproblems and conclusions from a set of common knowledge areas in addition to segment data and static reference data.

The proposed overall architecture is a blackboard model with multiple *knowledge sources* (KS). The blackboard stores a hierarchical representation of segment associations. The multiple knowledge sources, one for each sub-problem identified, work together to create, break and modify associations. The blackboard will also contain information describing the underlying dynamic model, which describes a possible tactical picture, used by the knowledge sources. Several instances of these models will exist, corresponding to multiple possible tactical pictures, based on the available data.

The proposed representation of associations has four layers (see Fig. 4). The bottom layer consists of individual segments created by the signal follower system. The second layer consists of temporal groups which are segment groups of temporally associated segments (the second segment appears to be an extension or extrapolation of the earlier segment, that is, the later segment is around the same frequency and bearing). The third layer consists of segment groups from all other associations where temporal groups from the second layer can be treated as individual segments. The fourth layer consists of associations of third layer groups to form a track.

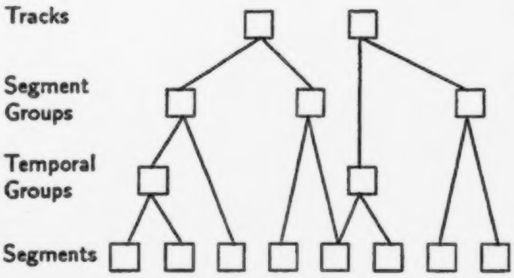


Figure 4: Four layer hierarchy for blackboard showing associations of segments.

The data record for a segment group should include the identity of the associated segments, a certainty measure which may be a fuzzy membership value, the hypothesized dynamic model and other assumptions used in reasoning by the knowledge source, and possibly the reasoning path used to conclude this association.

The blackboard is controlled by a process that focuses the attention of the knowledge sources and organizes the processing between them according to some prioritizing scheme. After each integration period any segments that have changed, either through lengthening or new identification information, can trigger the controller to re-evaluate their associations. The controller queries the knowledge sources as to whether the new information has an impact on the reasoning used to create an association. If there is an impact then the pair is put on the agenda along with any new pairs. The controller then organizes the agenda according to priorities and passes control to the appropriate knowledge source as each pair on the agenda is considered.

In addition to determining if new segment data has an impact on previous reasoning, the knowledge sources also generate, delete and modify the associations and the dynamic models.

The knowledge sources reason about structures on the blackboard using a rule base and information received from knowledge areas, which are processes that reason about concepts common to all knowledge sources (see next section). The rule base considers segments on the blackboard, temporal groups, segment groups and other non-acoustic information from a fact base, such as sensor characteristics. The knowledge sources also use historical information from past processing, such as associations, their certainty and the reason for a previous association. Any particular pair of segments is only ever considered by one knowledge source since a knowledge source is chosen based on which sensors created the segments and what sort of information it contains (BB or NB) thus the blackboard controller never needs to resolve conflicting conclusions from two knowledge sources. However, the knowledge sources themselves may need to resolve uncertain input data and inconsistencies between conclusions from different knowledge areas. Promising options for handling these uncertainties include Dempster-Shafer and fuzzy logic. Furthermore the knowledge sources must decide when to use an existing dynamic model rather than create a new one. The knowledge sources will attempt to make associations even if the segments are not yet ended.

The essence of the blackboard architecture described is a set of knowledge sources providing reasons for particular pairs of segment to be associated. New knowledge sources can easily be added to deal with specific reasons for groups of segments to be associated, with only minor changes to the controller and blackboard data structures.

Knowledge Sources

Knowledge sources are required for each subproblem. This section describes only two of the seven sources: NB to NB associations in the intrasensor case and BB to BB associations in the intersensor case. Knowledge sources read from and write to the blackboard.

Intra-sensor NB to NB Source

This knowledge source determines when two NB segments from the same sensor are from the same source and the reason why. Two NB segments could be received by the same sensor and be generated by the same source if the signal is travelling along multiple propagation paths, one segment is a harmonic or the signal was lost during some time interval. This knowledge source could deal with the first case by checking if the two segments were received from two different propagation paths, and if so they should also have similar pattern features. The segments should also have similar pattern features if one segment is a harmonic. The third possibility is a temporal variant.

Inter-sensor BB to BB Source

This knowledge source determines if two BB segments received by two different sonobuoys are from the same source. It must check if the geometry is reasonable with respect to target positions and maneuvering when the segments overlap in time and when there is a time lapse between the segments. Essentially the knowledge source reasoning tests the consistency of the information based on the hypothesis that one source is generating a BB signal and this signal is being received by two different sonobuoys, that is, do the bearings converge or diverge. If they diverge it might still be because of the propagation path. However, a sonobuoy usually receives signals along a direct path. This knowledge source might also check for similar patterns in the BB signal, such as an increase in *signal to noise ratio* (SNR), which, if present in both segments indicates the two segments are from the same source.

Knowledge Areas

Knowledge areas are processes that reason about particular concepts such as propagation paths, geometry and pattern features. Individual knowledge bases for reasoning within the various knowledge areas will use a wide variety of methods, including crisp rules, temporal predicates, case based reasoning, data base storage and retrieval, neural nets, fuzzy logic and uncertainty reasoning. We expect difficulties with normalizing the uncertainty measures across different knowledge areas. Knowledge Areas read from but do not write to the blackboard.

Fig. 5 illustrates the relationships between knowledge sources and the knowledge areas they use. Note that the target behaviour area is used directly by other knowledge areas, never directly by a knowledge source.

Similar patterns

This knowledge base is used to determine whether two segments have similar patterns based on prior knowledge (from knowledge elicitation or machine learning) about what features in a pattern are important. Similarity can be in one of three domains: frequency, bearing, or both. In addition, similarity in the context of temporal association implies similarities in slopes and continuity.

There are two types of similarity that can be ascertained. One is high-level similarity, based on the matching of features, and the second is low-level similarity, based on numerical measures of similarity. All methods will be adversely affected if the full time series of a segment is not available.

Three methods are proposed for determining high-level similarity. The first is fuzzy pattern-

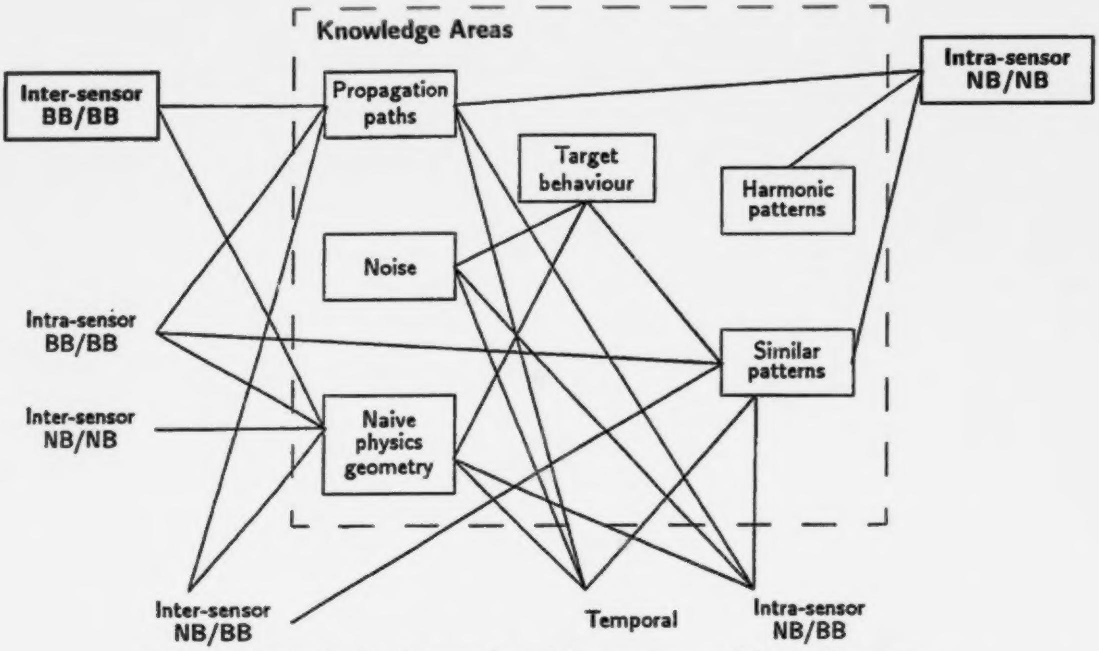


Figure 5: Relationship between knowledge sources and knowledge areas.

matching, either feature-by-feature, or by using common associations of features. The type of features would be the ones used by operators, for instance CPA, doppler shift, line width, and bifurcations. Individual feature values would be represented as linguistic variables (for instance "small Doppler-shift") whose membership functions are relative to measurable values. The similarity between two segments could be determined by the overlap between the fuzzy set memberships. The likelihood of the association of different features may depend on other knowledge about the source and the conditions.

This method also depends on a sub-process to identify features. Where these features have a simple numerical definition, this can be used. Otherwise, one possible method is to train a neural network to recognize them. The input would be a time series of pre-processed data, and the output a classification of a feature. Training would be provided by an experienced operator. Given the variety of time scales, a Time Delay Neural Network (TDNN) is a likely candidate. Note that feature identification is not required for some of the alternative methods described in this section. However, it may still be of use to other knowledge areas, such as target behaviour and naive physics/geometry.

A second method is to generate a low-order functional representation of one segment, based on the identified features and constrained by knowledge of target behaviour. If this function becomes a good predictor for the other segment, when similarly constrained, then it can be reasoned that they are causally linked.

A third method is to use an unsupervised neural network such as Adaptive Resonance Theory (ART), which finds its own features in the data.

Detecting low-level similarity is a numerical exercise, given that it does not rely on identifiable features or reasoning. One method is to train a neural network to reproduce an operator's determination of similarity, based on pre-processed data. A more conventional method is to perform a linear regression on the pairs of time series such as power, frequencies, SNR, with a number of different time shifts. If the two time series are highly correlated, they are likely to be from the same source, with the time shift indicating the difference in length of the propagation paths.

Harmonic patterns

This knowledge base is used to check whether two segments are harmonics. It looks for frequency multiples, and sums and differences of the frequencies of other segments. This determination is corroborated using current blackboard information about the identity and associations of the segments involved. This identification of harmonics should be consistent with what is known about similar patterns and with simple models of underwater acoustic signal propagation. It may use both forward and backward models of acoustics. The backward model starts with frequencies and reasons back to the properties of the acoustic source. The forward model starts with the presumed source and checks the expected harmonics for consistency with the data.

This knowledge area will generate hypotheses about "missing" segments, depending on the likelihood that the segment information on the blackboard is complete.

Naive physics/geometry

This knowledge base reasons about ship motion (both target and own ship) and ship position relative to a particular sensor in order to confirm or discount a model of the tactical situation. It could conclude that it is somewhat certain that only one ship caused the two segments under consideration from two sonobuoys since the bearings converge (possibly using information about propagation paths as well). Or it could conclude that it is very certain that two ships caused the two segments received by a towed array at different times because a ship could not travel fast enough to be in the right places to have generated both segments assuming some propagation path for each segment. This knowledge base will reason with the information received from the localization process.

This knowledge base will also identify a change of frequency as being consistent with the acceleration of a target in the direction of the signal propagation path, that is, it will conclude that a change in frequency is due to a doppler shift.

The knowledge representation structure most appropriate for this area is an IF THEN ... ELSE rule format combined with a simple model of motion. The reasoning should be backward chaining where the goal is asked by one of the knowledge sources for the subproblems.

Propagation paths

This knowledge base determines the possible propagation paths which a signal travelled. It could conclude that a segment was received from a bottom bounce, direct path, second convergence zone, etc. The input should include the current water conditions such as temperature, the weather (eg. raining, windy), the season, the location and any other parameter identified as useful for identifying cases. Two options are proposed: one rule-based, and the other case-based. The rule-based approach is most appropriate in familiar waters, where the acoustic conditions are well-known.

In the rule-based approach, the likely possible paths are determined on a heuristic model of known and presumed acoustic conditions. Some simple processing of bathymetric data features can be used to support the possibility of acoustic paths. Assignment of particular paths to segments can be based on features and bearing or beam of the segment. If some reliable measure of relative acoustic path length is available, simple geometric reasoning can be used to confirm multiple assignments of acoustic paths.

The most appropriate knowledge representation structure for unfamiliar waters is a data base of cases with a case based reasoning system. Rules are used to find similar cases and adapt the conclusion (if necessary) based on significant differences between the actual case and the database case.

Noise models

This knowledge base reasons about the various sources of noise and how they could interact with signals from a target (for example mask). It could conclude that there probably exists narrow band noise around frequency f. Or it could conclude a particular noise model.

A noise model is created based on input of the location, season, water temperature, depth of sensor and other non-acoustic data which can be used to specify a noise model consisting of:

- a statistical distribution model (Gaussian, white, etc);
- a trend (constant, transient, varying, etc); and
- a bias.

There are several knowledge representations and reasoning mechanisms appropriate to this problem. The noise model may be the result of a search through a data base. Case based reasoning or rule based reasoning could be used as well. Further knowledge elicitation is required before making a choice.

Another method for dealing with noise modeling may involve processing of the frequency time series data of a segment to extract noise from the signal using a technique such as a neural network. The output of the network would then be used as the new raw input for the knowledge source trying to associate two segments.

Target behaviour

This knowledge base reasons with reference knowledge about speed, maneuverability at different speeds, machinery noise and other target behaviour characteristics that affect segments. The reasoning adjusts particular parameters received from the reference knowledge according to some rules for the current situation. This knowledge includes general information applicable to all targets as well as information specific to certain classes of ships. The latter, more detailed information can be used based on hints from the classification process. The more detailed information can include knowledge about what maneuvers are possible but not likely, and about where they are likely to occur.

The knowledge representation scheme most appropriate to this area is a frame hierarchy where each frame consists of a knowledge base of rules, a list of properties and corresponding default values for the various aspects that describe target behaviours. A frame hierarchy includes progressively more specific knowledge about smaller and smaller classes of targets. The more is known about the identity of a target, the more narrow are the constraints on its behaviour. The knowledge area will not perform classification, but in the absence of such information may have to generate and test hypotheses about the class of target.

Input Data

This section briefly describes the structures of input data used by the knowledge areas. Fig. 6 illustrates which structures are used by each knowledge area.

Segment Data: Time series of frequency, bearing, SNR and classification information with a corresponding certainty (or fuzzy membership value)

Dynamic Model: Information describing a possible tactical situation, such as own ship position and velocity, target ship(s) position and velocity, bearings to possible target ships, etc.

Sensor Information: Sensitivity, position, orientation, depth and other non-acoustic information about a sensor.

Underwater Acoustic Conditions: Temperature, depth, layers, season, weather, location and other factors affecting underwater acoustic conditions.

Target Reference Data: Hierarchy of types, classes and specific identities of targets with details for each instance including maneuvering characteristics, maximum speed, narrow band signatures, broad band character, etc.

Conclusions

A *blackboard* model with multiple knowledge sources is proposed as an overall architecture. The blackboard would contain segment and segment pair information describing the underlying dynamic models (which describe the possible tactical pictures) used by the knowledge sources. Several instances of these models will exist, corresponding to multiple possible tactical pictures.

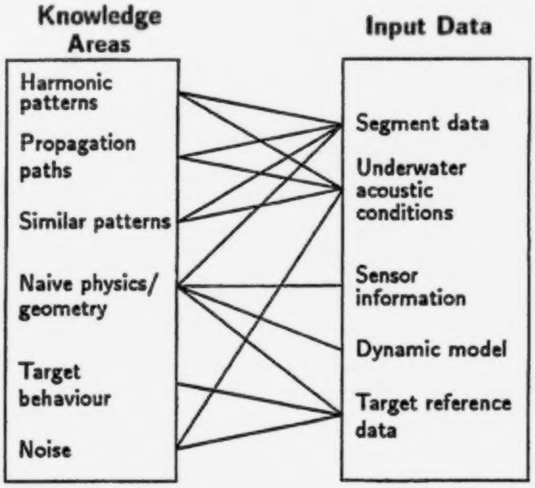


Figure 6: Knowledge areas and input data they use

The components of the system require *rule based reasoning*, *fuzzy logic* and *neural network* technologies. This is unlikely to be found in a single tool but using the blackboard model several tools can be integrated.

Although the blackboard does not require a way to handle uncertainties, all knowledge sources do as they must normalize the uncertainty measures that are returned by the knowledge areas. Problems are expected in normalizing the uncertainty measures across knowledge areas.

The success of the segment association process will greatly depend on the accuracy and completeness of its input data. Other processes in the system should share the blackboard and start providing tentative conclusions as soon as they are available.

References

[1] Iotek and Rockwood, "Expert Systems Study", Iotek Document No: DN0929 (Version 1.0), Oct 1992, and references therein.

YOCC 38

Kinematic Analysis and Controller Synthesis for Path Tracking in Automated Vehicles with Double Steering

Ahmad Hemami
Ecole Polytechnique
Montreal, Quebec H3C 3A7

#149750

Abstract

Double steering implies steering of a vehicle with both front and rear wheels. Its advantage over conventional steering is better performance of the vehicle in curve negotiation and its stability. In an automated vehicle by appropriate measurements of the orientation error and the offset from a desired path the controller unit or the auto-pilot is expected to bring the vehicle on its path when disturbed by road curvatures and by the road surface, wind and other perturbations. Mobile robots are within the class of low speed automated vehicles. For this class of vehicle, motion analysis and controller design for path tracking can be accomplished based on the kinematics of motion, because the effect of dynamics would appear as stable fast modes and may be neglected.

The synopsis of this study is the kinematics of vehicles with double steering and the design of an appropriate control logic for stabilization and improvement of their path following performance. The analysis is based on the non-holonomic constraint characteristics of the system. The equations governing the mechanics of path following is defined in the state space form. Four control schemes for such a vehicle are compared and the stability conditions are analyzed. The response of a real system is discussed and the conclusions are enhanced by simulation results.

Introduction

Double steering, all wheel steering, four wheel steering, whole wheel steering, dual-axle steering and fore-and-aft steering, all imply the class of vehicle steered by both front and real wheels. These vehicles offer a number of advantages in handling characteristics over the conventional vehicles with only front wheel steering. Lane change in high speeds and curve negotiation would be improved if the rear wheel is also steered according to an appropriate criteria based on the dynamic properties of a vehicle. For road vehicles this matter has been the subject of research and recently the various aspects of incorporating rear wheel steering has been investigated by a number of researchers. Mobile robots are slow speed vehicles that do not face the high speed stability and performance problems of road vehicles. On the other hand, they must work in the restricted areas, where many obstacles may be present, and follow relatively narrow corridors where a 90 degrees turn would be much facilitated by having a four wheel steering capability.

In the plane motion of a four wheel steering vehicle, if the steering wheels are free, that is, if no steering input is given to the vehicle, it possesses three degrees of freedom. In other words, such a vehicle can move independently in the longitudinal and lateral directions, and simultaneously rotate about an axis through its mass centre. These motions are governed by the applied forces to the vehicle from the outside. By fixing each set of the steering wheels at an angle, that is giving an input to the steering wheel, a degree of freedom is lost. This implies that if both the front and rear wheels are assumed a value the three motions are not independent any more .

Among the previous work on all wheel steering the references quoted here exemplify the investigations on the subject and the kind of problems studied. A comparison of rear wheel steering with front wheel steering is made by Whitehead using dynamics of a vehicle. In formulating the dynamics of a ground vehicle, from three (3) to eleven (11) degrees of freedom can be assumed, introducing the level of complexity according to the inclusion of roll and pitch motions, the dynamics of steering gears, tire reaction etc. Based on a three degrees of freedom dynamic model, Nisonger and Wormley [1979] study and compare the performance of single steering with double- (or dual-) steering for a number of cases that occur on a guideway, for a number of different control schemes. Nagi and Ohki [1989] use a special case involving only a lateral and yaw, two degrees of freedom, plane motion. (In their case, the rear wheels are steered in proportion to the steering angle of the front wheels, which is equivalent to loosing one degree of freedom; thus a total of two DOF). Based on the concept of virtual vehicle model following, they consider the theoretical performance of the transient response of a vehicle to disturbances like the side winds, etc.

Choosing an appropriate scheme for the control of the front and the rear wheels in a double steering system is another matter of investigation. For certain passenger cars schemes like proportionality of the rear steering angle to that of the front wheel, in the same or opposite direction, have been considered by car manufacturers [Nagai and Ohki]. The problem would be different, however, when there is a human operator and when an automated vehicle is considered. A comparative study of three different control schemes for the selection of the rear steering angle as a function of the front steering angle for a lane change manoeuvre is discussed by Sridhar and Hatwal[1992]. Other work in this respect is reported by Lee [1990, 1992]. A detailed study of the active controllers for a wide class of vehicle including the four wheel steering and a good collection of references for the related subjects can be found in the paper by Wickens[1991] and a control scheme is proposed in Mehrabi et al [1991].

In this paper, based on the kinematics of motion the control scheme for stabilization and error correction in path following is investigated. The results are applicable to mobile robots equipped with double steering, in view of their low operating speeds, and to any four wheel steering vehicle when operating at low speeds, whence the effect of dynamics become negligible. Four different feedback control schemes for the settings of the front and rear steering angles in terms of the two tracking errors, the heading error and the offset, are compared.

Kinematics of Motion

Figure 1 shows the schematic of a double steering vehicle and its configuration with respect to a desired path. At this moment the coordinate system XPY is attached to point P of the desired path, which is the nearest point on the path to the point C, the mass centre of the vehicle. PX is tangent to the path at this moment and PY is perpendicular to PX. Similarly, at point C the coordinate system UCW is attached to the vehicle, CU being the longitudinal axis and CW the lateral axis. The distance CP is the position error and the angle between PX and CU is the orientation error, denoted by ε_d and ε_θ, respectively. For determination of the two errors the same sign notion as the coordinate system can be adopted. In this sense, when C is to the left of PX the position error is positive; similarly when PX must be rotated counter-clockwise to become parallel to CU the orientation error is positive, otherwise negative.

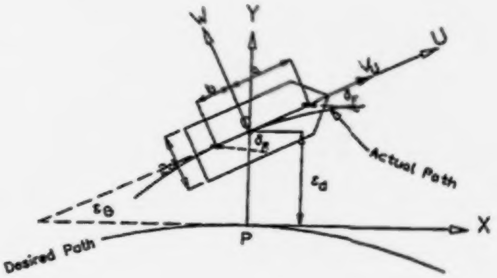


Fig. 1 - Status of a vehicle with respect to its desired path

In the representation of wheels in a four wheel steering system it is customary to represent each set of the front or rear wheels by a single wheel in the middle point of the corresponding axle. Obviously, the values for velocity, acceleration, etc. are the average of the values for the two wheels. This will be taken care in practice by the appropriate mechanism, such as a differential when it applies. The steering angles for the front and rear wheels are denoted by δ_F and δ_R, respectively.

The points of attachments for the front and rear tires are hereafter denoted by A and B, respectively, as depicted in figure 2a. This figure also shows the vectors of linear velocity for these points, which based on the non-holonomic constraint assumption are along the direction of the associated wheels, making angles δ_F and δ_R with the longitudinal axis. *a* and *b* are the distances of A and B from C, respectively.

When the vehicle moves, the front tire rolls along a circle whose centre is on a line normal to V_A and the rear tire rolls along a circle whose centre is on a line normal to V_B. As a result, point S, the intersection of a normal to V_A and a normal to V_B, is the centre for rotation of the vehicle at this instant. The projection of S on BA is denoted by M. M is the only point on the longitudinal axis which has a purely forward velocity. The forward velocity of the vehicle, thus, may be represented by the velocity of this point, V_M. The vehicle has an instantaneous angular velocity ω obtained from

$$\omega = \frac{V_M}{\overline{MS}} \qquad (1)$$

where the sign of $\overline{MS}$ determines whether ω is positive or negative. The velocities of the other points on BA have a lateral component equivalent to a rotation by ω about point M, as depicted in figure 2b. In this sense,

$$V_A = V_M + \overline{MA}\,\omega \qquad (2)$$

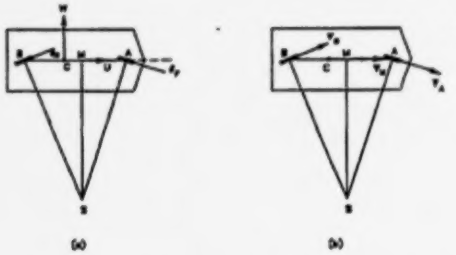


Fig. 2 Instantaneous centre for rotation

and

$$V_B = V_M + \overline{MB}\,\omega \quad . \tag{3}$$

The coordinates of point S can be determined from the geometry solving simultaneously the equations for liens SA and SB. SA is perpendicular to the front wheel and passes through A, thus its equation in the UCW coordinates is

$$w = -(u-a)/\tan \delta_F \tag{4}$$

Similarly, the equation of line SB is

$$w = -(u+b)/\tan \delta_R \tag{5}$$

Solving equations (4) and (5) yields

$$\overline{CM} = \frac{a \tan \delta_R + b \tan \delta_F}{\tan \delta_R - \tan \delta_F}$$

$$\overline{MS} = \frac{a + b}{\tan \delta_F - \tan \delta_R} \quad . \tag{6}$$

It is evident from equation (6) that when $\tan \delta_F = \tan \delta_R$, that is when the front and rear wheels are parallel, then the vehicle moves on a straight line along the direction of the wheels. In any other case the vehicle moves on a circle with the centre at S; the radius for point M is MS, which is minimum when δ_F and δ_R assume their maximum absolute values and have different signs. Equation (6) also indicates that if

$$\frac{\tan \delta_F}{\tan \delta_R} = \frac{-a}{b}$$

then point M will coincide with point C and the mass centre C has a purely forward velocity (zero side slip angle).

Moreover, it may be seen by investigation of the values of *CM* and *MS* that among δ_F and δ_R whichever has a larger absolute value is dominant. That is to say, it governs the direction to which the vehicle turns (right or left). For instance, when abs(δ_F) > abs(δ_R), if the front wheel is to the right the vehicle will turn right no matter whether the rear wheel is to the right or left. The effect of the rear wheel in such a case when the front wheel is kept constant at certain angle is only to make variation in the radius and the centre of rotation.

It follows from equations (1) and (6) that the angular velocity ω may be defined in terms of the steering angles of the wheels in the form of

$$\omega = \frac{V_M}{a+b}(\tan \delta_F - \tan \delta_R) \quad . \tag{7}$$

It also follows from equation (6) that if it is desired that the mass centre of a vehicle moves on a circle whose centre in the body attached coordinates UCW is given, the steering angles must be set to

$$\tan \delta_F = \frac{a - S_U}{S_W}$$

$$\tan \delta_R = -\frac{b + S_U}{S_U} \tag{8}$$

where S_U and S_W are the coordinates of the centre of the circular path with respect to the current position of the vehicle. Equation (8) may be used for path planning purposes.

When a vehicle moves, if the steering angles are not exactly as required by the path two errors will be generated. Figures 3a,b show these errors after a small motion of the vehicle for the period Δt. According to figure 3-a

$$\epsilon_\theta = \Delta\theta \pm \Delta\psi \tag{9}$$

where Δψ represents the effect of curvature change in the path. Δψ can be positive or negative; this is reflected by the inclusion of a ± sign in equation (9).

Noting that the angular velocity is the rate of change of orientation it follows from equation (9) that

$$\omega = \dot{\theta} = \frac{\Delta\theta}{\Delta t} = \dot{\epsilon}_\theta \pm \dot{\psi} \quad . \tag{10}$$

Moreover, the lateral displacement depends on the forward displacement and the orientation change. In this sense, the rate of the lateral displacement of point

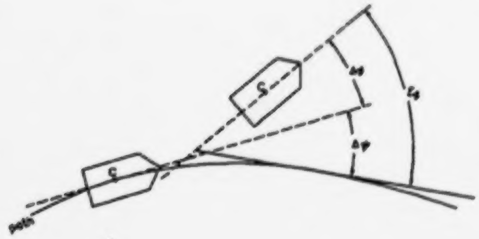


Fig. 3a orientation error generated

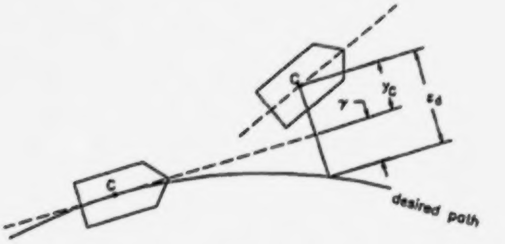


Fig. 3b position error generated

M in the XPY coordinates is

$$\dot{y}_M = \dot{w}_M = V_M \Delta\theta \tag{11}$$

where V_M is the forward velocity. For point C, the rate of lateral displacement, therefore, is given by

$$\dot{y}_C = \dot{y}_M + \overline{MC}\,\omega = V_M \Delta\theta + \overline{MC}\,\omega \tag{12}$$

Substituting from equation (9) in equation (12) leads to

$$\dot{y}_C = V_M e_\theta + \overline{MC}\,\omega \pm V_M \Delta\psi \tag{13}$$

Figure 3-b reveals the relationship between the position error introduced and the lateral displacement as a vehicle moves, thus

$$e_d = y_C \pm \gamma \tag{14}$$

where γ represents the effect of curvature change in the path; it can be positive or negative. It then follows from equations (13) and (14) that

$$\dot{e}_d = V_M e_\theta + \overline{MC}\,\omega \pm \dot{\lambda} \tag{15}$$

where λ denotes all the terms representing the effect of the path curvature. In the case a straight line is to be followed this term and ψ in equation (10) are zero.

Controller Synthesis

The equations of motion in the state space form can be derived from equations (7), (10) and (15) as follows:

$$\begin{bmatrix}\dot{\varepsilon}_d\\ \dot{\varepsilon}_\theta\end{bmatrix} = \begin{bmatrix}0 & V_M\\ 0 & 0\end{bmatrix}\begin{bmatrix}\varepsilon_d\\ \varepsilon_\theta\end{bmatrix} + \frac{V_M}{a+b}\begin{bmatrix}b & a\\ 1 & -1\end{bmatrix}\begin{bmatrix}\tan\delta_F\\ \tan\delta_R\end{bmatrix} \pm \begin{bmatrix}\dot{\lambda}\\ \dot{\psi}\end{bmatrix} . \tag{16}$$

The open loop system has two poles at zero and is unstable. The system can be stabilized by selecting an appropriate feedback law to determine the two inputs, $\tan\delta_F$ and $\tan\delta_R$. Equation (16) is in the general form

$$\dot{x} = A\,x + B\,u + d \quad . \tag{17}$$

The vector d can be regarded as a disturbance if it is not known; it can be taken care by a feed-forward action if sufficient information about it is available, for example if the path is a circle.

The control problem is to define a feedback matrix K such that at each instant based on the measured values of the two errors, in position and orientation, the values for the steering angles of the front and rear wheels are determined. In this sense,

$$\begin{bmatrix}\tan\delta_F\\ \tan\delta_R\end{bmatrix} = \begin{bmatrix}k_1 & k_2\\ k_3 & k_4\end{bmatrix}\begin{bmatrix}e_d\\ e_\theta\end{bmatrix} \tag{18}$$

and the closed-loop plant matrix is given by

$$A + B\,K = \frac{V_M}{a+b}\begin{bmatrix}ak_3 + bk_1 & ak_4 + bk_2 + a+b\\ k_1 - k_3 & k_2 - k_4\end{bmatrix} . \tag{19}$$

In order to find the feedback matrix K, first, consider the two extreme cases, when only ε_d exists and when only ε_θ is present, as shown in figure 4. In the former, it is not necessary to change the vehicle orientation; for this, the front and rear wheels must be parallel, that is, theoretically δ_F and δ_R must be equal. In the latter, by the same token, δ_F and δ_R must have opposite signs so that the vehicle traces a curve with a small radius, so that the orientation correction takes place faster. The feedback matrix K has the structure

$$K = \begin{bmatrix}k_1 & k_2\\ k_1 & -k_2\end{bmatrix} \tag{20}$$

where k_1 and k_2 are negative. On the other hand, in the light of the fact that the mechanism for steering allows larger steering angle for the front wheel than the rear wheel, it can be assumed that the feedback matrix is in the form of

$$K = \begin{bmatrix} k_1 & k_2 \\ \frac{k_1}{m} & \frac{-k_2}{m} \end{bmatrix} \tag{21}$$

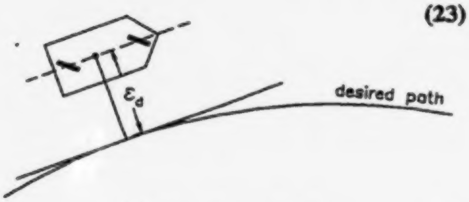
where m > 1. This is more realistic from practical point of view. With this assumption, in a vehicle the front wheel is always dominant, as was discussed earlier.

Furthermore, equation (16) reveals that ε_d is influenced by ε_θ whereas ε_θ is not affected by ε_d. For this, it is appropriate to give more weight to ε_θ in the feedback matrix. The structure of K, in such a case, is in the following form

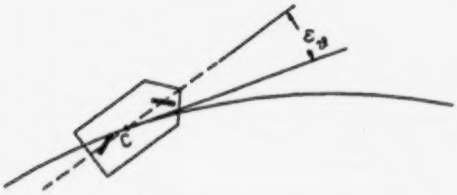
$$K = \begin{bmatrix} k_1 & k_2 \\ \frac{k_1}{m} & \frac{-k_2}{m} \end{bmatrix} = k \begin{bmatrix} 1 & p \\ \frac{1}{m} & \frac{-p}{m} \end{bmatrix} \tag{22}$$

where p > 1 and k < 0. With such a structure for matrix K, the closed-loop plant matrix of equation (19) will be

$$A + BK = \frac{kV_M}{a+b} \begin{bmatrix} \frac{a}{m} + b & \frac{-ap}{m} + bp + \frac{a+b}{k} \\ 1 - \frac{1}{m} & p(\frac{1}{m} + 1) \end{bmatrix} . \tag{23}$$



(a) position error only



(b) orientation error only

Fig.4 Methods of steering for error correction

The performance and stability of the controlled system depend on the eigenvalues of the closed-loop plant matrix, which are functions of the parameters m and p, the gain k and the forward velocity of a vehicle. The forward velocity has a direct effect on the eigenvalues, which in turn determine how fast the two errors are corrected. This governs the rate of change of the required steering inputs δ_F and δ_R. From practical point of view and the limitation of the mechanical system of the wheels steering the larger the forward velocity is the smaller the gain k must be.

The two parameters m and p and the loop gain k affect the stability of the controlled system. There are two stability conditions that indicate the relationships among these three values in order for the system to be stable. The first conditions for stability is:

$$(b+\frac{a}{m}) + p(1+\frac{1}{m}) > 0 \tag{24}$$

In view of the fact that both m and p are greater than unity and that a and b are positive values, this condition is always true. The second condition reduces to the following form:

$$k < \frac{m-1}{2p} \tag{25}$$

This condition is also always satisfied in light of the fact that m > 1 and k has a negative value.

Alternative Control Schemes

Although the feedback matrix K in equation (22) is possibly a perfect choice for the cases when only one error exists, as it can be seen by inspection the front wheel steering angle is not always larger (in absolute value) than the rear steering angle. That is to say, it does not assume m times the rear steering angle for all the values of the two errors ε_d and ε_θ. To make sure this requirement is always met, an alternative control law may be employed which adjusts the magnitude for the rear steering angle to be always smaller in absolute value than the steering angle of the front wheel. In the following policy δ_F is always determined according to equation (22), that is:

$$\tan \delta_F = k(\varepsilon_d + p\varepsilon_\theta) \tag{26}$$

but δ_R is calculated based on the value of δ_1:

$$\tan \delta_1 = \frac{k}{m}(\varepsilon_d - p\varepsilon_\theta) \tag{27}$$

as follows:

$$\delta_R = sign(\delta_1)\begin{cases} abs(\delta_1) & \text{if } abs(\delta_F)/m > abs(\delta_1) \\ abs(\frac{\delta_F}{m}) & \text{if } abs(\delta_F)/m < abs(\delta_1) \end{cases} . \tag{28}$$

Selecting δ_R as in equation (27) is equivalent to having a varying feedback matrix K which, depending on the values of ε_d and ε_θ, assumes a form as given in equation (22) or one of the following:

$$(a) \qquad K = k\begin{bmatrix} 1 & p \\ \frac{1}{m} & \frac{p}{m} \end{bmatrix} \tag{29}$$

when δ_1 is positive and $abs(\delta_F)/m < abs(\delta_1)$, or

$$(b) \qquad K = k\begin{bmatrix} 1 & p \\ \frac{-1}{m} & \frac{-p}{m} \end{bmatrix} \tag{30}$$

when δ_1 is negative and $abs(\delta_F)/m < abs(\delta_1)$.

A stability analysis for these two cases indicates that the stability conditions for case (a) are that $p > 0$ and $k < 0$; these two conditions are always satisfied. For case (b) one condition is that $k < 0$ and the other becomes:

$$p > \frac{a - bm}{m + 1} = a - \frac{m}{m + 1}(a + b) \quad . \tag{31}$$

This condition can be always satisfied for a particular system. It can be concluded that a system controlled based on the control laws (26) - (28) does not go to an unstable condition.

It is to be remarked that the control matrix in equation (29) corresponds also to a controller scheme in which the steering angle of the rear wheel is always proportional to that of the front wheel and both wheels steer to the same side. Similarly, the control matrix in equation (30) correspond to such a controller but when the front and rear wheels steer to opposite directions.

Parameter Variations Studies

In what follows the effect of the variation of the three parameters m, p and k in the feedback control matrix on the eigenvalues of the closed-loop plant matrix controlled according to the control policies in equations (22), (29) and (30) are studied. Also for a set of nominal values for these parameters the behaviour of the controlled system is simulated and examined. For a real system, addressed in reference [4], the variations of the closed-loop poles with the variation of each of these parameters are plotted in figures 5, 6 and 7. For the vehicle under study a = 0.32 m, b = 39 m. The nominal values for m,p and k are 4, 2 and -0.75, respectively; V_M is such that $kV_M = -0.5$.

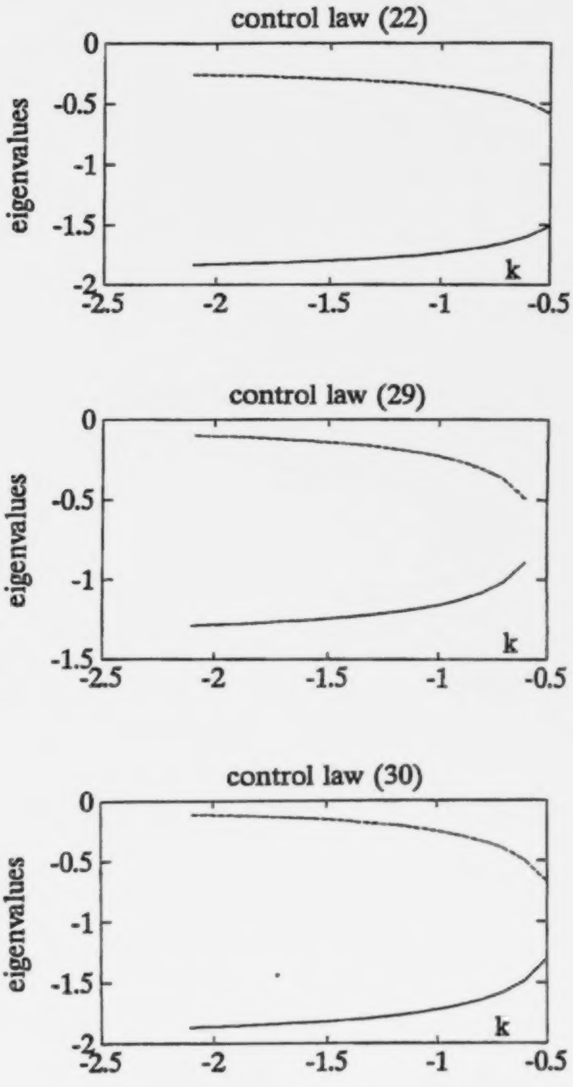
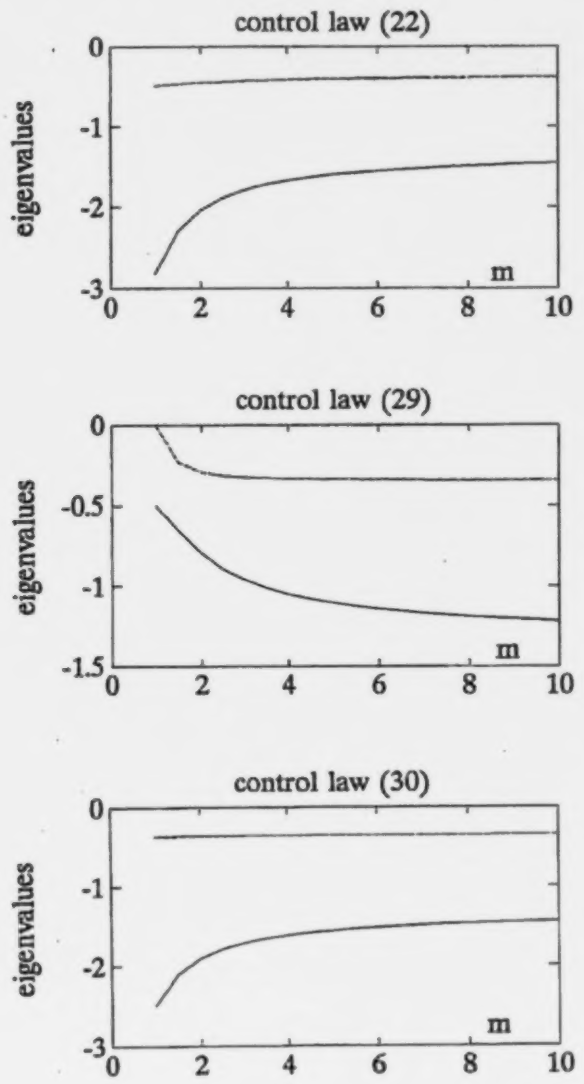


Fig. 5- variation of k

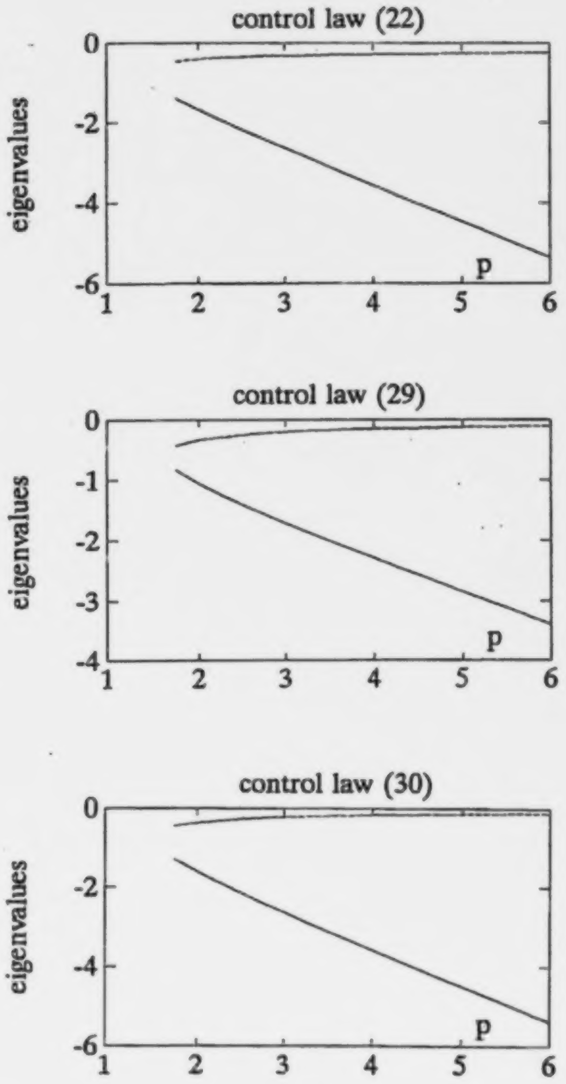
The results illustrated in figures 5-7 depend on the physical dimensions of a vehicle; in this particular example, the closed-loop poles are all real. Similar


control law (22)

control law (29)

control law (30)

Fig. 6- variation of m


control law (22)

control law (29)

control law (30)

Fig. 7- variation of p

relative results can be obtained for other vehicles. The conclusions then can be generalized, as far as the numerical values of parameters are not considered (This is shown by another example not presented here).

The studies made indicate that the variation of the parameters do not have a significant effect on the location of the (dominant) closed-loop poles in all the three control policies considered in equations (22), (29) and (30), as far as they are not near their critical ranges. This implies that under the effect of these control laws the controlled system is not sensitive to parameter variations. In the system considered here, which one of the poles is always dominant, only for the control law (29) parameter m must not approach its minimum value of unity. A general conclusion is that large values for parameter p and high gains (more negative values) are undesirable. However, increasing m does not lead to instability (A theoretical value of ∞ for m is equivalent to having front steering only).

Error Correction Performance Studies

From the above results it could be concluded that as far as the system stability is concerned no advantage could be found between the various control laws discussed. As for comparison of the performance in path tracing, we want to compare the behaviour of a system when controlled according to (a): feedback in equations (22), (b): equations (26)-(28), (c): when the rear wheel is steered proportional to the front wheel in the same direction (Eq. 29) or (d): in the opposite direction (Eq. 30), respectively. since the behaviour of such a system, as formulated in this study, depends on the location of the closed-loop poles, first these poles are compared. Corresponding to m = 4, p = 2, g = -0.5 and V_m = 1 m/sec, for case (a) the system poles are:

$$\lambda_1 = -0.584 \ , \ \lambda_2 = -1.508 \qquad (32)$$

For the second scheme they assume one of the following values:

$$\begin{aligned} &\lambda_{1,2} = -0.694 \pm 0.217i \\ &\lambda_1 = -0.675 \ , \ \lambda_2 = -1.304 \end{aligned} \qquad (33)$$

depending on the magnitudes of the errors, as discussed earlier. The two sets of values in equation (33) correspond also to the cases of controller (c) and (d), respectively.

As it can be seen, no significant difference can be found between the location of the poles. This implies that no significant preference may be attributed to any of the four schemes. For all cases the controlled system has a performance more or less similar to that shown in figure 8. The simulation is performed for a straight path with an initial position error of 0.5m and orientation error of 0.2 rad. Since the forward velocity affects the behaviour of the system, figure 8 is scaled based on the distance travelled by a rather than time.

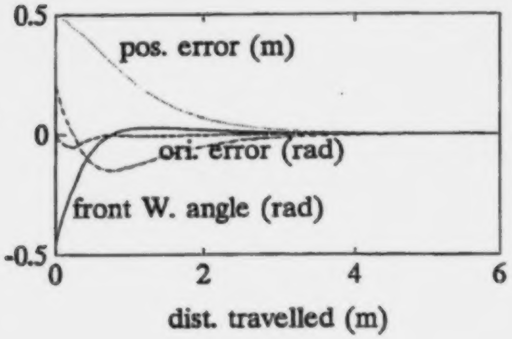


Fig. 8 - Error correction Performance

Summary

Kinematic analysis is carried out for the class of vehicle equipped with double steering. The state space equations governing the position and orientation errors in a path following task are formulated. Four feedback control schemes for adjusting the steering wheels are compared. The stability of the controlled system and the effect of the variation of the control parameters are investigated. The performance of a vehicle in following a straight line, when implementing these control laws, is simulated. The study reveals that for the slow speed vehicles for which the kinematic relations would suffice, no significant preference could be found for any of the control schemes considered.

References

1. Lee, A.Y., Design of Stability Augmentation Systems for Automotive Vehicles, J. of Dynamic Systems, Measurement and Control, 112 (1990), pp 489-495.
2. Lee, A.Y., A Preview Steering Autopilot Control Algorithm for Four-Wheel-Steering Passenger Vehicles, J. Dynamic Systems, Measurement and Control, 114 (1992), pp 401-408.
3. Mehrabi, M.G., Cheng, R.M.H. and Hemami, A., Control of a Wheeled Mobile Robot with Double Steering, Proc. IROS 91: IEEE Int. Workshop on Intelligent Robots and Systems, Osaka, Japan, Nov. 1991, pp 806-810.
4. Mehrabi, M.G., Cheng, R.M.H. and Hemami, A., Dynamic Modelling and Control of Wheeled Mobile Robots, Theory and Experiment, To be published in the Proc. 2nd IEEE Conf. on Control Applications, Vancouver, Sept. 1993.
5. Nisonger, R.L. and Wormley, D.N., Dynamic Performance of Automated Guideway Transit Vehicles with Dual Axle Steering, IEEE Trans. on Vehicular Technology, VT-28 (1979), pp 88-94.
6. Nagai, M. and Ohki, M., Theoretical Study on Active Four-Wheel-Steering System by Virtual Vehicle Model Following Control, Int. J. Vehicle Design, 10 (1989), pp 16-33.
7. Sridhar J. and Hatwal H., A comparative Study of Four Wheel Steering Models Using the Inverse Solution, Vehicle System Dynamics, 21 (1992), pp 1-18.
8. Whitehead, J.C., Rear Wheel Steering Dynamics Compared to Front Steering, J. Dynamic Systems Measurement and Control, 112 (1990), pp 88-93.
9. Whickens, A.H. Dynamics of Actively Guided Vehicles, Vehicle Dynamic Systems, 20 (1991), pp 219-242.

90C 204

Basic ROMAC Control Algorithms

Bart Hall
B Hall Industries
146 McAllister Rd., North York, Ontario M3H 2N7

Julius J. Grodski
Defence and Civil Institute of Environmental Medicine
PO Box 2000, Downsview, Ontario, Canada, M3M 3B9

#149751

Abstract

RObotic Muscle ACtuator (ROMAC) is a pneumatically operated, single acting, compliant actuator which has been the subject of control algorithm research. ROMAC presents a unique control problem because there are two antagonistically operating actuators for a single mechanical degree of freedom joint, as opposed to usual robotic actuation schemes where a single actuator, like an electric motor, operates a single mechanical degree of freedom joint. An intuitive approach to algorithm design was used because of the system non-linearities and the antagonistic actuator configuration. The combination of system non-linearities gives the joint the uniquely advantageous qualities of independently-controllable position and compliance. Such a joint is analogous to a human joint like an elbow actuated by a biceps/triceps antagonistic pair. Control algorithms for single link configurations were tested. A control algorithm called Pressure Ratio control (PR) maps required actuator pressures to combinations of joint position and compliance. Position feedback (PS) control operates quickly and accurately but has stability problems.

Introduction

Robotic Muscle ACtuator (ROMAC) has provided a challenge in efforts to control ROMAC actuated joints. ROMAC is a pneumatically operated, single acting, compliant actuator[1]. Two antagonistically coupled actuators are typically used to operate a single degree of freedom rotary joint. This is a unique control problem because of the antagonistic operation of a single mechanical degree of freedom joint, as opposed to the usual case of one dual-acting actuator, such as a dc motor, per joint. Pneumatic operation of the ROMACs combined with non-linearities in the joint and ROMACs result in the uniquely advantageous qualities of independently-controllable position and compliance. A joint operated by antagonistic ROMACs is analogous to a human joint, such as an elbow actuated by a biceps/triceps muscle pair.

The objective of the development of the control algorithms is to attain rapid, accurate joint motion while utilizing the unique joint compliance capabilities offered by ROMACS. Both of the control algorithms in this paper were discussed previously[2,3] from a more general perspective. Additional activity has provided more practical insight into their behaviour. The fundamental control problem is to strategically operate the pneumatic supply to each ROMAC in the antagonistic pair. Internal actuator pressures and joint angle are used as measured quantities to control one proportionally acting inflate/exhaust servo valve per ROMAC. The pressure and tension developed by each actuator depend on integration of gas flow, which is in turn dependent on supply or atmospheric pressure, internal ROMAC pressure and the valve aperture commanded. ROMAC tension is proportional to pressure, but both ROMAC internal volume and tension are non-linear functions of actuator stroke. With the usual rotary joint configuration where the ROMACs are connected to lever arms, actuator strokes are also a non-linear function of joint angle. These non-linearities and system principles make the control issue challenging. As a result an intuitive approach to algorithm design was used as opposed to extensive analysis and simulation. Most control knowledge is intended for linear systems and hence not easily applied in this case.

There is an area of control algorithm research which deals with both positional accuracy and compliance or force control issues. This work can further be divided by type of actuation used. ROMAC is typified by being pneumatically driven (vs. electric or hydraulic) and by being used in at least a pair-configuration antagonistically because it is single acting in tension only. Related research by others includes pneumatic servoing of cylinders and algorithms for antagonistically actuated systems.

Moore et al.[4] and Pu et al.[5] discuss point to point position servoing of a double acting pneumatic cylinder and the problems caused by the compressibility of air. These include "time delays, low stiffness, little natural damping and complex non-linearities"[4]. In reality, a double acting pneumatic cylinder is a compressively-antagonistic system with two piston plus cylinder units coupled by a common stroke. Key similarities between a pneumatic cylinder and a ROMAC actuated joint are common actuator strokes and actuator volume changes caused by stroking. A difference is the added non-linearity of ROMAC volume with stroke as compared to a pneumatic cylinder. An advantageous difference is the lack of sticking friction in a ROMAC compared to a pneumatic cylinder. Ivlev[6] resorted to Teflon cylinder seals and the Utah/M.I.T. Dextrous Hand[7,8,9] used custom graphite and glass cylinders to avoid the sticking friction problem. Moore et al.[4], Pu et al.[5], Ivlev[6] and Noritsugu[10,11] were concerned primarily with accurate positioning of a cylinder, generally regarding stiffness as something which reduced position accuracy while overcoming friction or load. The most directly related work to ROMAC control algorithms is the servoing of the Utah/M.I.T. Dextrous Hand as it deals with positioning and force control for its joints. It also uses a pair of single acting pneumatic cylinders creating antagonistic tensions to operate each joint. At this point a basic difference of approach occurred. The Utah/M.I.T. Dextrous Hand uses custom electromechanical pressure feedback servovalves (the valves have proportional filling/exhausting elements as part of their internals) to make each cylinder act as a low impedance tension source. The ROMAC system uses a proportionally acting spool valve with all feedback methods done via digital control loops, allowing greater flexibility in determining actuator behaviour. The different approaches in conjunction with some unique properties of ROMACs have lead to different control algorithms.

The basic ROMAC test bed consists of an arm actuated by an antagonistic pair of ROMACs as shown in Figure 1. Each ROMAC is instrumented with a pressure sensor and the joint has a position encoder. Each ROMAC has its own variable aperture fill/exhaust valve. Analog outputs from the pressure transducers go through an analog-to-digital (A/D) conversion into the digital control computer which sends out a valve command through the digital-to-analog (D/A) converter. The encoder counter circuit is read directly into the computer. Thus, control algorithms are implemented in digital form in software allowing for easy testing and verification.

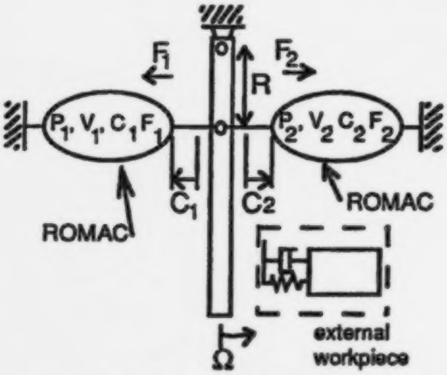


FIGURE 1. Diagram of a Basic Antagonistic ROMAC-Actuated Joint

System Model

The following is a model for the single joint with two antagonistic ROMACs shown in Figure 1. Some attempts at simplification are shown in an effort to get an intuitive understanding of the system in linear terms. The first equations are shown as simplified forms for modeling and are based on data from experimental characterization of a 4" 4x2 arcuate ROMAC.

The actuator volume has a slight pressure bulge (shown by the data at 0.8" contraction: 215 cc @ 5 psig, 230 cc @ 60 psig, 240 cc @ 120 psig) and an ln(contraction) dependence which has been linearised at 0.8" for modeling purposes in equation (1). The 0.8" operating point is conveniently chosen as it occurrs at the centered joint position. The ROMACs are set up with 0.8" contraction when the joint is centered so the full 1.6" stroke of both actuators can be used.

$$V\,(@0.8'') = 230 + 100\,C \qquad (1)$$

where:

V is volume in cc
C is contraction in inches

The basic actuator force model including some error at contraction extremes and low pressures is equation (2). Equation (3) has a PxC non-linearity which actually hides the strong exponential dependence of force on contraction.

$$F = 22.8 - 3.12\,P + 18.5\,P\exp(-C) \qquad (2)$$

$$F\,@0.8'' = 22.8 + 11.9\,P - 8.33\,P\,C \qquad (3)$$

where:

F is force in lbs
P is pressure in psig
C is contraction in inches

Equation (4) is the differential equation for mass gas flow through the servo valve in or out of the ROMAC using ideal gas and adiabatic assumptions.

$dm/dt = (V/(RT))\ (dP/dt) + (P/(RT))\ (dV/dt)$ (4)

where:

V is ROMAC internal volume
P is ROMAC internal pressure
R is the universal gas constant
T is temperature

The mass gas flow dm/dt has four cases depending on whether the flow is sonic or subsonic and in or out of the ROMAC. Sonic flow occurs when $P_{high}/P_{low} > 0.5285$ (absolute pressures) and subsonic flow when $P_{high}/P_{low} < 0.5285$. The servo valve responds so that a positive signal for aperture fills the ROMAC and a negative signal exhausts it to the atmosphere. The $f_{subsonic}()$ is a complicated equation not included in equation (5) to simplify the modeling of the system for the purpose of general illustration. It is only here in equation (5) that the controllable input to the system, valve aperture, appears. Most operation is subsonic.

Four dm/dt cases: (5)

i) $dm/dt_{(sonic\ fill)} = k\ P_s\ A$
ii) $dm/dt_{(subsonic\ fill)} = k\ P_s\ A\ f_{subsonic}(P_s, P_r)$
iii) $dm/dt_{(sonic\ exhaust)} = k\ P_r\ A$
iv) $dm/dt_{(subsonic\ exhst)} = k\ P\ A\ f_{subsonic}(P_r, P_{atm})$

where:

k is a flow constant for the specific plumbing
P_s is the high pressure of the air supply
P_r is ROMAC internal pressure
A is the area of the valve aperture
P_{atm} is atmospheric pressure

The following listing of equations (6) to (19) is presented in a top-down order of increasing detail to allow one to see the complexity and coupling of variables which would occur when substitutions are performed to create a single system differential equation from them.

$J\ D^2(\Omega) = T_2 - T_1$ (6)

where:

J is moment of inertia of the link
D^2 is the second derivative operator
Ω is joint angle as in Figure 1
T_1 and T_2 are the torques from the ROMACs

Calculating torques with a slight geometry simplification:

$T_1 = F_1 * \text{effective radius arm}_1 = F_1\ R\cos(\Omega)$ (7)
$T_2 = F_2 * \text{effective radius arm}_2 = F_2\ R\cos(-\Omega)$ (8)

Substituting in (6) and simplifying:

$J\ D^2(\Omega) = R\cos(\Omega)\ (F_2 - F_1)$ (9)

From equation (3):

$F_1 = 22.8 + 11.9\ P_1 - 8.33\ P_1\ C_1$ (10)
$F_2 = 22.8 + 11.9\ P_2 - 8.33\ P_2\ C_2$ (11)

Calculating contraction from Figure 1 with a slight geometry simplification:

$C_1 = 0.8 - R\sin(\Omega)$ (12)
$C_2 = 0.8 + R\sin(\Omega)$ (13)

Applying equation (4):

$P_1 = (RT / (dV_1/dt))\ (dm_1/dt - (V_1/RT)\ dP_1/dt)$ (14)
$P_2 = (RT / (dV_2/dt))\ (dm_2/dt - (V_2/RT)\ dP_2/dt)$ (15)

Applying equation (1):

$V_1 = 230 + 100\ C_1$ (16)
$V_2 = 230 + 100\ C_2$ (17)

For the sake of brevity:

dm_1/dt = appropriate substitutions in eqn. (5) (18)
dm_2/dt = appropriate substitutions in eqn. (5) (19)

The purpose of listing the components of the system model is to illustrate the non-linearity of the system. In particular, linearized modelling coefficients would all change significantly if a different operating point was used. The gas flow could possibly be approximated as a first order time constant based on most of the flow being subsonic, except that the time constant changes by at least a factor of two during joint motion because the internal ROMAC volume changes that much. Not only would the rate of growth of pressure from gas flow change because of the change in cavity volume, but the sealed-in mass of gas would also change pressure from a change in cavity volume. The pressure would no longer represent the net effect of a history of gas flow, contradicting the expected behaviour of a 1st order accumulator type of process. Another non-linearity is the exponential increase of tension-per-pressure transduction gain as the ROMAC extends. Less dramatic non-linearities are the trigonometric functions of Ω in equations (7), (8), (9), (12) and (13) for a +/- 40° range of Ω. The pressure dependent volume bulge has been neglected in equations (16) and (17), and the slight force-stroke hysteresis has not been modeled in equations (10) and (11).

The basic observations possible about the overall system are that the transduction of servo valve aperture command to actuator tension is an integrating process, and that the number and severity of the non-linearities prevents the creation of a simple linear time invariant system model suitable for application of standard control theory. Standard control theory also falls far short in dealing with redundant actuation as it usually deals with single dual-acting actuation.

An interesting observation about the system is that there are two degrees of freedom in the actuation of a single motion degree of freedom joint. One could conceive of the joint as having two state variables, one being the joint position and the other being the internal antagonistic tension between the ROMACs. The antagonistic tension is significantly related to the inherent joint stiffness as will be illustrated later with Figure 2.

From the view point of control algorithms, the system description could be regarded as a black box with inputs, natural behavior and outputs. Outputs could be physical measurements provided by sensors such as actuator pressures, joint angles and the non-instrumented interaction forces between the robotic link and an external object. The inputs to the system could be relatively constant air supply pressure, constant atmospheric pressure, valve aperture commands and external force/position disturbances to the ROMAC actuated link.

An initial objective in control was to attain satisfactory joint motion with no external influences. With observation of the inputs and outputs it can be seen that the only controller generated inputs are the valve aperture commands. They are the only means of accomplishing rapid damped position control of what is essentially a second order spring-mass system. A second objective of control was to take advantage of the compliance of the ROMACs. The basic target feature was to be able to set the inherent joint stiffness independently of the joint position. The level of inherent joint stiffness chosen functions as an open loop type of control of the forces developed between the link and a work piece.

Pressure Ratio (PR) Control

The first and most basic control method natural for a ROMAC actuated joint is Pressure Ratio (PR) control because it utilizes the unique inherent ROMAC force-contraction-pressure (fcp) characteristic. The methodology for PR control is based on the graph shown in Figure 2. It is this unique fcp characteristic which gives a clear control advantage for ROMACs. PR control may be very undamped under extreme circumstances of very large inertial loads, but it will always be unconditionally stable. I.e. it will *never* go unstable, in contrast to position control for a pneumatic cylinder. Position control of a pneumatic cylinder goes progressively unstable as its positional stiffness is raised by increasing the feedback gain. In contrast, a ROMAC joint's stiffness is limited by the maximum pressure the actuator can withstand.

PR control operates as follows. A pressure sensor on each ROMAC with a proportional gain of fill/exhaust aperture setting per psig of pressure error creates an individual control feedback loop per ROMAC. In this way the pressure in each ROMAC can be set with 0.5 psig accuracy and a response time of 0.1 to 0.2 seconds for a massive command like $\Delta P = 60$ psig. The essence of the control then lies in choosing the pressure for each actuator in a meaningful way. One can see in Figure 2 a set of crossed curves of tension for each ROMAC in an antagonistic pair. Excluding external joint loading, a choice of pressure for each ROMAC results in a choice of operating pressure curves which have an equilibrium intersection point. At that point the antagonistic actuator tensions are equal and a stable joint position results.

One can see that there are several possible pressure combinations for a joint position. In Figure 2, 10 psig and 30 psig pressures result in approximately +12° joint angle, as does 50 psig and 130 psig. The pair of lower pressures results in a lower joint stiffness (higher compliance; compliance = 1/stiffness) and the pair of higher pressures results in higher joint stiffness. The joint stiffness is illustrated in Figure 2 by noting the size of reaction force developed in response to an angular deflection. By following the two constant pressure force lines, one can see that one actuator decreases its force while the other increases its force resulting in a Δforce for the Δangle. At the higher pressures, the Δforces are larger, resulting in higher stiffness. Thus an independent setting of spring constant type stiffness can result from choice of pressures at the same joint angle.

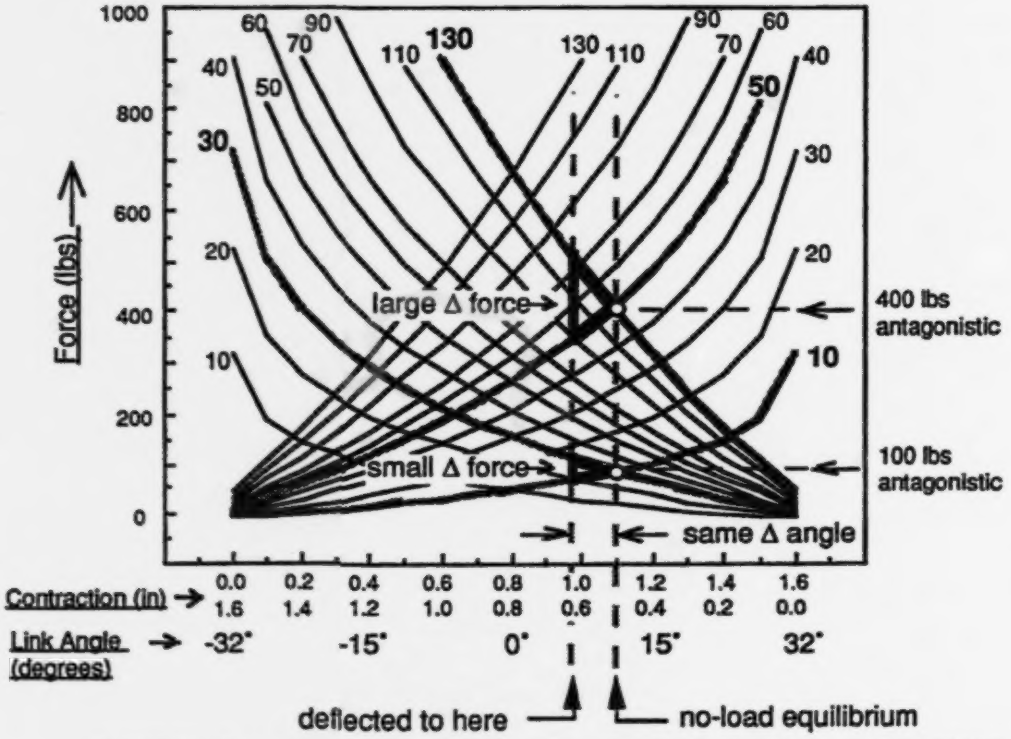


FIGURE 2: Crossed Constant Pressure Force Curves for a Pair of Antagonistic ROMACs

The question of choosing a pair of actuator pressures was originally solved as indicated by the name Pressure Ratio control. Theory and experiment show that at a fixed contraction (actuator length) the force produced is proportional to the gauge or differential pressure across the casing membrane. Hence at fixed joint angle, doubling the pressure in each actuator should simply double the equilibrium antagonistic force and not change the joint angle, resulting in the joint angle being defined by the ratio of the pressures. In actual practice this was not the case. The actuator force has a mechanical spring component from the flexure of the casing and a slight force hysteresis as a function of stroke. An experimental no-load mapping procedure was followed where one actuator pressure was held constant and the other reduced in steps with the resultant joint positions recorded. Oscillations were induced prior to recording each position to minimize the effects of the hysteresis. The data should have shown a unique mapping of ratio of gauge pressures to joint angle but instead it showed different ratios giving equal angles. The effect was not scattered, but uniform. Plotting a modified ratio $(P_1+3)/(P_2+3)$ vs joint angle coerced all the experimental ratios onto a single function of joint angle. This modified pressure ratio as a function of angle was adopted as an angle predicting "mapping" equation to help in choosing pressures for PR control.

A second experimentally-based mapping equation of joint stiffness is also required since use of joint position as the only input for the modified ratio equation results in an indeterminate pair of actuator pressures. Stiffness measurement was done by striking an impulse to the joint to induce a measured oscillation at various pressure combinations and joint angles. The periods of oscillation from which the stiffnesses were calculated were found to be amplitude dependent. Therefore representative mapping from a standardized choice of amplitude was done. An equation of stiffness as a function of the higher actuator pressure and joint angle was derived by separating the variables into graphical plots which had clear dependencies recognized. Then the relationships from the plots were combined into a full mathematical equation. Thus a stiffness and position could define one actuator pressure through the stiffness equation and the other actuator pressure could then be determined by the modified pressure ratio equation. The experimentally mapped choice of pressures could then be individually servoed for each ROMAC.

Plain PR control works well, being rapid and stable, but suffers from a basic air-spring / mass natural oscillation tendency. It was determined that a "velocity" or "damping" contribution to the control could be attained by noting the following. In a typical control problem, velocity feedback to an actuator whose force is proportional to its input will damp the position response. Since in a ROMAC system the actuator force is proportional to the integral of the input signal (valve aperture), an additional differentiation of velocity (= acceleration) is required to get the regular "velocity" damping feedback effect. An acceleration signal was obtained from double differentiating the joint position and the signal was added to the aperture command created by the proportional pressure error servo loop. The desired damping effect was achieved resulting in much better control. Figure 3 shows the improvement.

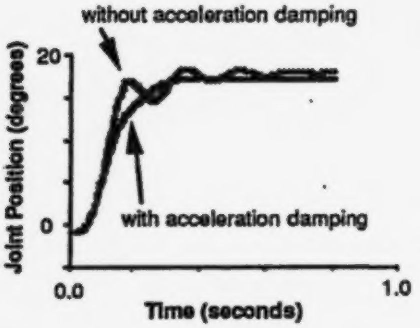


Figure 3. Example Position Step Responses for PR Control

Its worth note that the behaviour of the joint in a step position command is not truly representative of the control behaviour. The step response for a linear system is representative, but the highly non-linear nature of a ROMAC actuated joint changes the effective settings of constant controller gains as functions of position within the range of joint rotation, average pressure levels in the actuators, supply pressure, direction of loading or travel, amplitude of position step, attainment of saturation levels in the system, etc. Experimental data showed some positional, directional and saturation effects in a moderately comprehensive set of step response tests too voluminous to report here. The oscillation shown in Figure 3 actually had only about 3% overshoot but was clearly oscillatory. Increased joint inertia produced a more typical overshoot oscillation waveform.

The basic behavior of PR control is to configure the joint as a torsional spring with slight damping, variable spring rate and variable initial zero load rotation angle. The primary drawbacks of PR control are that joint loading always produces sizable position deflections and PR's no-load positional accuracy isn't very good. The inaccuracy is due to 1) reliance on a set of experimental data which is still expected to be appropriate in the face of likely small mechanical adjustments to the joint, 2) reliance on the quality of the fit of the mapping equations to the data, 3) friction effects in the joint and 4) force-stroke hysteresis in the actuators. In a joint configured for 70 degrees of travel with older prototypes, a pair of pressures could result in a position anywhere inside a 2 to 3 degree window, depending on the direction of approach and direction of any loading just experienced. A source of hysteresis on the older prototypes is movement of the cable harness relative to the casing during contraction. This became apparent when audible slip and stop noises were noticed under some operating conditions. Although this hysteresis exists, it should be noted that it does not cause sticking of the joint motion as does the hysteresis of a pneumatic cylinder.

Position (PS) Control

PS control was developed as a means to improve the accuracy of no-load and loaded joint position control. Several characteristics result from proportional feedback of joint position error to logically fill or exhaust each ROMAC (i.e. valve signal proportional to position error) to increase or decrease its pressure and tension. The natural action of integrating the gas flow results in a basic integral feedback type of response with tendency to oscillation, an automatic equalizing response to external joint loading and good positional accuracy. The overall response is rapid despite reliance on integrating action because the pneumatic flow integrating action rate is high. Operation of PS without acceleration feedback results in extreme instability. Acceleration feedback gives marginal stability with reduced line supply pressure that can easily be upset by an increase of line supply pressure, increased joint inertia or slight increase of feedback gain.

In principle the behavior of PS could be regarded as infinite joint stiffness with an approximated first order time constant response. This is practically limited by the saturation limits of zero and maximum pressure for the actuator, and is not "infinite" in magnitude of force response. A problem with PS control is the indeterminate actuator pressures attained during operation. A central joint location can arbitrarily result in high or low pressures, leading to varied performance. In general, PS control could be made stable by either limiting link inertia or by lowering the gain to the point of degrading the performance when larger inertias were used. An alternative for stabilizing is to pre-limit the valve signals to less than full valve capacity so that the valves behave as if they had a smaller maximum flow capacity. A similar degradation of performance occurred when more stabilizing was attained from smaller flows. In order to create presentable example step responses, PS control with acceleration feedback and the d/dt(position error) signal (which will be covered in the next section) was operated at optimal tunings for speed and stability for inertial loading which was varied by a factor of six in Figure 4.

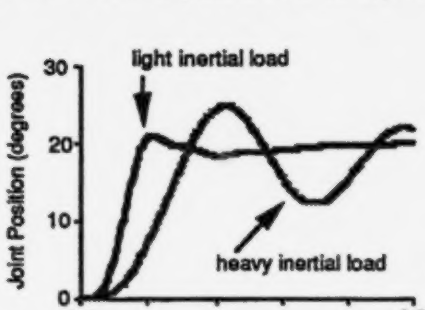


Figure 4. Example Position Step Responses for PS Control

Additional Basic Control Signals

Some additional signals were tried in relation to PR and PS control. Derivative of pressure error signal was considered for PR, but the servo loop was already very fast and stable without it. It could not be readily ascertained whether the nature of the pressure sensor placement might already be introducing some derivative-like action by venturi or ram-air type effects. Integral of pressure error signal was implemented at one time in an attempt to improve the positional accuracy of PR, but it was found that force-stroke hysteresis was the real problem. At one point when very early actuator prototypes developed leakage problems, integral gain effectively compensated for the resulting pressure errors. A similar accuracy effect was noted with integral of position error gain applied to PS in the face of air leakage. Without leakage, integral of position error gain on PS simply caused the joint to slowly oscillate around the target position. Derivative of position error signal applied to PS had remarkable effect. The lag in following a ramped position was reduced and the damping was improved. This was clearly evident in its ability to stabilize pure PS control to the point of useability. One might notice that proportional position error gain almost behaves as if the motive force was proportional to the generated valve signal, instead of its real dependence on gas flow integration. Given this observation, one can apply the regular rationalization for the effect of derivative of position error feedback. When the position error is increasing, more signal for motive force is created (reducing lag or reducing overshoot) and when the error is decreasing reduction of the signal for motive force occurs (reducing overshoot).

There are side effects to the joint stiffness from acceleration feedback and derivative of position error feedback. Acceleration feedback reduces the signal for motive force in the direction of the acceleration which slows down actuator motivated motion but also "slows down" motion resulting from an external motivating force, i.e. it adds a force to resist the external force. Derivative of position error feedback kicks in extra signal for motive force to resist an increasing position error, whether the increasing error is caused by a rapidly ramping command position, inertially induced position overshoot or velocity of joint motion resulting from an external force or object. As a result, both signals result in an increase in the component of joint stiffness that is velocity or motion dependent. Previous mention of joint stiffness was not explicit about time dependency. Basic PR stiffness is essentially independent of reasonably limited speed of joint deflection because of the speed of the pressure servo loops. Basic PS stiffness is very high with slight time delay of actuator pressure change, but acceleration and derivative feedback effects make the stiffness in response to more rapid deflections higher.

Conclusion

A pragmatic approach has been used for development of joint control for antagonistic ROMAC actuated joints based on ROMAC pressure sensing and joint angle sensing. Experimental and analytical characterization of the features of such a joint showed severe non-linearities. In addition to how relatively little studied the field of controlling antagonistic actuation is, control of forces and positional accuracy is a very recent development. These problems precluded easy application of a standard control design solution. The resultant pragmatically designed controls contributed to a fundamental understanding of the system and successfully exploited the unique ROMAC fcp characteristics. A broader view of joint control, beyond only speed and positional accuracy, includes stability and stiffness considerations. The inherent moderate stiffness of pneumatics and the unique ROMAC PR control in particular offers a simple, effective alternative for contact sensing followed by selection of pure position or force servoing.

Results from tests of new control algorithms combining PR and PS will soon be available. Current research also includes incorporation of ROMAC tension sensing in control algorithms with results to be available shortly.

References

1. G. B. Immega and J. J. Grodski, "RObotic Muscle ACtuator (ROMAC) Robotic Applications", Proc. DREP/RRMC Military Robotic Applications Workshop, Victoria, B.C., RRMC Special Report 87-1, pp. 49-54, 1987.

2. J. J. Grodski andG. B. Immega, "RObotic Muscle ACtuator (ROMAC) Theory", proc. DREP/RRMC Military Robotic Applications Workshop, Victoria, B.C., RRMC Special Report 87-1, pp. 41-48, 1987.

3. G. B. Immega, "ROMAC Muscle Powered Robots", RI-SME Conference on Robotics and Research, Scottsdale, Arizona, 18-21 Aug 86; Paper Ms 96-777

4. P.R. Moore, R. H. Weston, T. W. Thatcher, "Compensation in Pneumatically Actuated Servomechanisms", Transactions of the Institute of Measurement and Control, Vol. 7, No. 5, Oct.-Dec. 1985.

5. J. Pu, R. H. Weston, "Motion Control of Pneumatic Drives", Microprocessors and Microsystems, Vol. 12, No. 7, September 1988.

6. V. I. Ivlev, "Economical Pneumatic Servodrive for a Robot Arm", Stanki i Instrument, Vol. 54, Issue 9, pp. 7-8, 1983.

7. S. C. Jacobsen, H. Ko, E. C. Iversen, C. C. Davis, "Control Strategies for Tendon-Driven Manipulators", IEEE Conference on Robotics and Automation, Scottsdale, Arizona, May 14-19, 1989.

8. K. B. Biggers, S. C. Jacobsen, G. E. Gerpheide, "Low Level Control of the UTAH/M.I.T. Dextrous Hand", IEEE International Conference on Robotics and Automation, San Francisco, California, April 14-17, 1986.

9. S. C. Jacobsen, J. E. Wood, D. F. Knutti, R. T. Johnson, K. B. Biggers, "Design of the UTAH/M.I.T. Dextrous Hand", IEEE International Conference on Robotics and Automation, San Francisco, California, April 7-10, 1986.

10. T. Noritsugu, "Development of PWM Mode Electro-Pneumatic Servomechanism Part I: Speed Control of a Pneumatic Cylinder", The Journal of Fluid Control, Vol. 17, #1, 1987.

11. T. Noritsugu, "Development of PWM Mode Electro-Pneumatic Servomechanism Part II: Position Control of a Pneumatic Cylinder", The Journal of Fluid Control, Vol. 17, #1, 1987.

90C203

Trajectoires Optimales de Robot en Utilisant le Concept d'Isotropie

Jean Poulin, Martin Lortie et Paul Gely
Ecole de technologie supérieure
4750 Henri-Julien, Montréal, Québec, H2T 2C8.

#149753

Sommaire

Une trajectoire optimale de robot ne peut pas s'obtenir simplement par digitalisation ou par apprentissage surtout si le robot est redondant. Par contre, la programmation hors de la cellule robotisée, permet d'aborder de façon approfondie une telle optimisation. Cet article présente la recherche d'une trajectoire optimale associée à une tâche en utilisant le principe de l'isotropie. Les premiers développements concernant cette optimisation ont été intégrés à une application particulière de l'industrie aéronautique.

Introduction

De plus en plus, la programmation hors-ligne est devenue un atout important pour les industries qui veulent augmenter la flexibilité et la productivité de leurs robots.

Ce type de programmation présente, en effet, plusieurs avantages tels qu'expliqué par P. Gely en 1992 [1]. Aussi, elle peut être utilisée pour diverses applications industrielles. Dans l'application décrite ici, le robot déplace une buse pour vaporiser une barrière thermique de céramique sur des coques axisymétriques en rotation. Le calcul d'une telle trajectoire considère les multiples risques de collision ainsi que les contraintes rattachées à la fois aux limites physiques du robot et à ses redondances. D'autres contraintes, imposées par le procédé de vaporisation, sont également prises en compte telles que la perpendicularité du jet de vaporisation de la buse par rapport à la surface à vaporiser et la distance libre entre la buse et cette surface.

Conventionnellement, la trajectoire du robot est enseignée directement sur le site d'utilisation. Dans ce cas, l'enseignement point par point du trajet monopolise la cellule du robot durant une longue période. De plus, une telle pratique n'assure guère de précision dans le trajet obtenu. L'opérateur peut difficilement, par exemple, évaluer l'orientation de l'axe de l'effecteur pour chaque configuration à enregistrer.

Par surcroît, si une collision ou une sortie articulaire se produit durant la digitalisation, toute la procédure d'enregistrement doit être recommencée à un nouveau positionnement pour la pièce. Aussi, l'enseignement point par point réalisé par l'opérateur ne permet pas d'optimiser la trajectoire en fonction de l'accessibilité de la pièce, la flexibilité et la vitesse du robot. L'opérateur, même habile au maniement du robot, travaille de façon aveugle sans exploiter toutes les capacités du robot.

Grâce à la programmation hors-ligne, des paramètres du procédé et de la dynamique du robot peuvent être considérés et pondérés de façon simultanée.

Si la génération hors-ligne de la trajectoire tient compte seulement des limites articulaires et de l'environnement de travail du robot, le long de cette trajectoire, le robot ne travaillera pas forcément avec souplesse. A proximité de ses singularités, la trajectoire comportera des soubresauts, inacceptables pour une trajectoire continue telle que souhaitée. Il devient alors intéressant de toujours conserver le robot dans les configurations présentant le plus de contrôlabilité possible (i.e. loin des singularités) tout le long du trajet. La programmation hors-ligne doit donc inclure dans ses calculs un critère représentant la contrôlabilité des mouvements du robot tout au long de la future trajectoire. Celà est d'autant plus vrai lorsqu'il est impossible d'accéder à l'algorithme de contrôle du robot.

Cet article présente un critère de contrôlabilité, soit le principe de l'isotropie. Des résultats sont illustrés à travers une application industrielle.

Concept d'isotropie

En respectant déjà toutes les contraintes reliées au robot, à la cellule et au procédé implanté, l'obtention de la trajectoire finale peut s'appuyer sur plusieurs critères de notre choix. Un de ces critères pourrait être, par exemple, d'assurer des variations minimales à chacune des articulations du robot entre chaque configuration successives. On pourrait ainsi s'attendre à ce que les mouvements du robot le long de la trajectoire soient réguliers et sans à coup. Or, ce n'est pas nécessairement le cas.

En effet, une approche aussi simple de recherche de trajectoire n'évite pas la convergence vers les singularités sans que la proximité de celles-ci ne soit signalée. Malgré l'ajout du critère minimisant les variations aux articulations, des mouvements subits devront être générés pour éviter ces singularités.

Le critère de l'isotropie permet de prévenir ce genre de problème car il permet de maintenir le robot dans une configuration la plus proche possible de la configuration idéale de l'isotropie (i.e. loin des singularités). Dans le contexte de la robotique, l'isotropie est une caractéristique qui peut être attribuée à une configuration particulière: un robot est dans une configuration isotropique lorsqu'il peut se déplacer avec une contrôlabilité égale (*"isos"*) dans toutes les directions (*"tropos"*). Ce concept évalue donc la manipulabilité d'un robot dans une configuration donnée.

Lorsque le robot s'éloigne des configurations isotropiques, il perd de son agilité à se déplacer dans certaines directions. Finalement, lorsque le robot se trouve sur une singularité, les mouvements deviennent complètement bloqués dans certaines directions: le robot perd des degrés de liberté. En définitive, une configuration isotropique fait opposition à une configuration singulière.

Théorie

Idéalement, un robot se trouve dans une configuration isotropique lorsqu'il est loin de ses singularités qui ne peuvent être détectées qu'à partir du Jacobien.

Le Jacobien, matrice reliant les vitesses des diverses articulations avec la vitesse au bout de l'effecteur, permet d'établir les équations dynamiques du mouvement. Cette matrice permet également de connaître les forces et les moments à chaque articulation à partir de la force et du moment connus au bout de l'effecteur. La matrice du Jacobien, notée **J**, se décrit comme suit:

$$\begin{bmatrix} v \\ \Omega \end{bmatrix}_{bout\ outil} = J\,[\dot{\theta}_i] \qquad (1)$$

où v et Ω sont respectivement les vitesses linéaires et angulaires à l'extrémité de l'effecteur et,
$\dot{\Theta}_i$ est la vitesse angulaire dans le cas où toutes les articulations du robot sont rotatives.

Pour un robot connu, une singularité est détectable lorsque la matrice Jacobienne **J** est singulière, c'est-à-dire qu'elle n'est pas inversible. Alors, il n'y a pas de solution $\dot{\Theta}_i$ telle que le vecteur vitesse $[v\ \Omega]^T$ peut-être satisfait.

Il est clair qu'une configuration singulière est indésirable et doit être évitée à tout prix. Mais on doit également être prudent dans les zones avoisinantes de cette singularité car la contrôlabilité se trouve graduellement affectée. Cette proximité de singularité peut être cernée en utilisant les valeurs singulières de la matrice **J** qui sont obtenues de la décomposition en valeurs singulières. Brièvement, la matrice **J** peut être décomposée telle que:

$$J = U[\sigma]V^T \qquad (2)$$

où

$$[\sigma] = \begin{bmatrix} \sigma_1 & 0 & 0 & 0 & \dots & 0 \\ 0 & \sigma_2 & 0 & 0 & \dots & 0 \\ 0 & 0 & \sigma_3 & 0 & \dots & 0 \\ 0 & 0 & 0 & \sigma_4 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \dots & \sigma_m \end{bmatrix} \qquad (3)$$

et où $\sigma_1 \geq \sigma_2 \geq \sigma_3 \geq \dots \geq \sigma_m \geq 0$ sont les valeurs singulières (m étant le nombre d'articulations du robot).

La contrôlabilité peut ainsi être évaluée par le produit suivant proposé par T. Yoshikawa en 1984 [2]:

$$Cr = \sigma_1 \cdot \sigma_2 \cdot \sigma_3 \cdot \sigma_4 \cdot \ldots \cdot \sigma_m \qquad (4)$$

Plus ce produit Cr est élevé, plus le robot est dans une configuration contrôlable (loin des singularités) et lorsque Cr tend vers 0, la configuration s'approche alors de la singularité. Cr est, en fait, un indice de proximité de la configuration singulière la plus proche.

Autres Critères d'Évaluation Considérés

La recherche d'une solution optimale ne peut se faire en utilisant seulement le principe de l'isotropie. D'autres contraintes liées au robot comme au procédé doivent être considérés. En incluant l'isotropie, cinq critères d'évaluation sont retenus pour l'optimisation:

1) L'isotropie
2) Les limites des articulations du robot:
   Il faut éviter, pour chaque solution, que les articulations approchent et atteignent leurs limites;
3) L'inclinaison de l'outil:
   Autant que possible, dans notre application, il est souhaitable que l'axe de l'outil soit perpendiculaire à la surface de la pièce;
4) La similitude entre les angles articulaires de deux configurations consécutives:
   Lors de la génération du parcours, celà évite de retenir des solutions trop différentes entre deux configurations consécutives;
5) La rotation de l'extrémité de l'outil entre deux configurations consécutives:
   Ce critère joue un rôle semblable au précédent mais il agit de façon plus spécifique sur la rotation de l'outil.

A chacun de ces cinq critères, une pénalité normalisée est calculée selon une fonction continue. Ces différentes pénalités obtenues peuvent être ensuite pondérées selon l'importance allouée au critère associé. La somme pondérée des pénalités des cinq critères donne la pénalité globale de la solution.

Description de l'application et simulation

Le logiciel de simulation et de génération de trajectoires est utilisé pour un procédé de vaporisation au plasma. Un robot vaporise de la céramique sur une coque axisymétrique (figure 1) en rotation. D'autre part, pour respecter les exigences du procédé, plusieurs paramètres doivent être pris en compte pour générer cette trajectoire dont la distance libre entre la buse et la pièce.

Dans le logiciel de simulation, après avoir introduit le profil de la coque avec les points d'approche et de retrait, la recherche de la trajectoire optimale est lancée d'après les deux étapes suivantes:

1) Etude de l'accessibilité à la coque et optimisation des solutions:

Une étude d'accessibilité sur tout le pourtour de la coque est réalisée tel qu'illustrée à la figure 1 où "p" profils à "n" configurations chacun, sont posés équidistants sur le pourtour. Pour chacune de ces configurations, on doit rechercher une solution idéale qui évite toute collision.

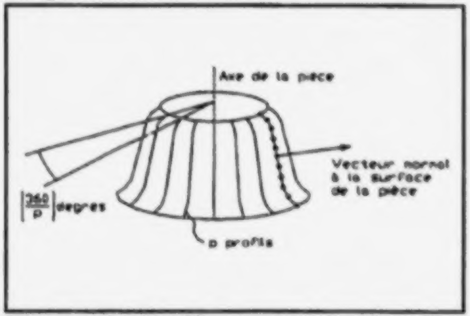


Figure 1: Profils autour de la pièce axisymétrique.

Puisque la buse n'est pas indexée autour de son axe ni même par rapport à la normale à la pièce, plusieurs solutions de positionnement sont envisageables telles que le montre la figure 2. Sur cette figure, la buse peut être inclinée par rapport à la normale d'un angle α et se situer autour de celle-ci selon l'angle β. De plus, par l'angle Φ, la buse peut tourner autour de son axe. Donc, pour chaque configuration "i", plusieurs solutions sont possibles. Les calculs d'optimisation permettent de trouver les valeurs de α, β et Φ donnant la plus faible pénalité en regard des cinq critères énumérés précédemment en incluant l'isotropie. Une simple méthode de gradient est utilisée pour effectuer cette recherche dans l'espace.

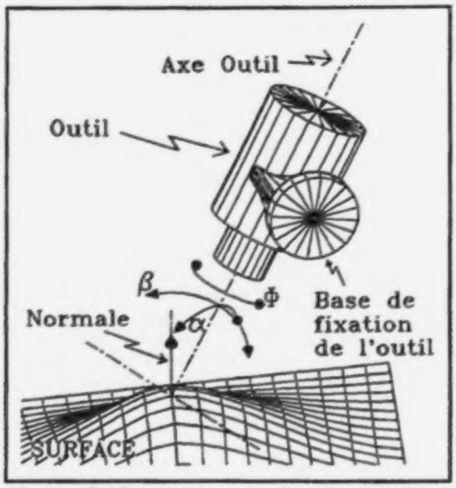


Figure 2: Paramètres α, β et Φ décrivant le positionnement de l'effecteur à un point donnée de la surface.

2) Recherche d'un parcours:

A partir des données de l'étude d'accessibilité précédente, la recherche du parcours optimal est lancée en reliant des configurations consécutives. Le logiciel choisit et classe les solutions retenues selon les variations articulaires.

Résultats

Une étude d'accessibilité a été réalisée sur une pièce dont le profil est présenté à la figure 3. Chaque vecteur normal à la courbe désigne une position "i" sur les "n" possibles qui ont été étudiées sur le profil.

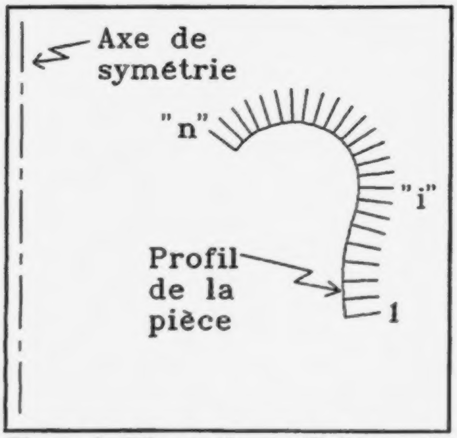


Figure 3: Pièce typique utilisée pour la simulation

La figure 4 illustre les résultats obtenus à partir de ce profil. Les courbes de variation des articulations 3 et 5 sont tracées en considérant (Iso.) et en excluant (Std.) le critère de l'isotropie.

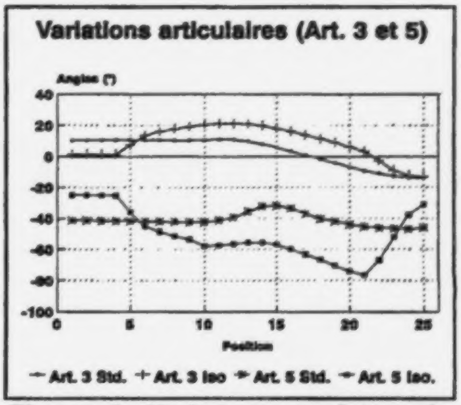


Figure 4: Variations angulaires des articulations 3 et 5 en fonction de la position.

Les courbes obtenues montrent que les variations angulaires sont plus importantes lorsque le critère de l'isotropie est pris en compte. Ces variations, quoique plus importantes, présentent un avantage majeur car elles permettent d'éviter toute proximité de singularités présente lorsque le critère de l'isotropie est ignoré.

Discussion

Le logiciel de simulation, fonctionnel depuis un an, n'incluait pas jusqu'alors le critère de l'isotropie dans la recherche de la solution optimale. Les trajectoires obtenues, à ce stade du développement, présentaient parfois des anomalies comme de subites variations angulaires proches de singularités.

Les premiers résultats obtenus suite à l'inclusion du critère de l'isotropie s'avèrent très prometteurs car on constate des mouvements plus continus du robot. Ce critère apporte une vision plus globale durant la recherche des solutions car il prévient les configurations singulières et on peut ainsi les contourner. Ce critère fort simple devient très important à considérer lorsque l'algorithme de contrôle du robot ne peut être utilisé facilement pour générer les trajectoires.

Conclusions

La programmation hors-ligne d'une trajectoire de robot est un problème complexe qui ne se limite pas au seules contraintes physiques du robot et de la cellule. La morphologie et la dynamique du robot doivent être prises en compte. Des critères additionnels doivent être ajoutées pour prévenir les inconvénients liés à cette morphologie et cette dynamique. Par exemple, durant une tâche de vaporisation, il est inconcevable que le robot ait des mouvements subits ou se heurte à des singularités. La trajectoire doit impérativement être continue (i.e. exempte de singularités).

Après d'autres recherches afin d'obtenir des trajectoires continues, le critère de l'isotropie s'avère un critère additionnel très intéressant. Il permet de situer le robot par rapport à ses points de redondance. Son intégration dans les algorithmes de recherche des solutions (lorsque l'algorithme de contrôle du robot est inconnu) est un apport précieux car à tout moment il situe le robot par rapport à ses régions critiques. Ainsi donc, des décisions de contournement peuvent être anticipées afin de minimiser les inconvénients rattachés à ces régions critiques et produire ainsi une trajectoire valide.

Références

1. **Gely, P.**, "Robotique: Avantages de la génération hors ligne de trajectoires et de leurs simulations", *Conférence Canadienne et Exposition*, IAIA, 1992

2. **Yoshikawa T.**, "Analysis And Control Of Robot Manipulators With Redundancy", *Robotics Research*, MIT Press, Cambridge, 1984.

901660

Finding Neural Network Weights Without Training

Dr. L. Jenkins
Department of Engineering Management
Royal Military College of Canada
Kingston, Ontario, K7K 5L0

Major S. Jones
Department of Electrical Engineering
Royal Military College of Canada

#149755

Abstract

Artificial neural networks (ANN) are now accepted as an effective tool for many classification tasks. Developing an ANN for a particular task involves determining the best architecture (number of layers, number of neurons in each layer, connections between neurons and transfer function at each neuron) and training the ANN in order to decide the weight on each connection. Though there are some guidelines to help select a suitable architecture, the selection still involves much trial and error. Determining the weights can require many trial computer runs, each of which can take literally days, with no guarantee of when or if success will be achieved.

Nearly 30 years ago a formula was specified for classification between different groups on the basis of each group's value on various attributes and the weights were determined very efficiently by the simplex algorithm of linear programming. Recently a number of researchers have explored methods of using such an approach to determine rapidly (mere seconds of digital computing) the best architecture and the best weights for an ANN to perform the classification task. More recently an induction algorithm based on an information-theoretic measure has been proposed to determine quickly and unequivocally the best architecture and weights in an ANN for classification.

The paper describes briefly these two methods and illustrates how the output of each optimization is used to define the architecture and weights of an ANN for the classification task. The approaches are compared with using a back-propagation ANN learning algorithm for the same classification.

1. Introduction

Over the last decade, artificial neural networks (ANN) have established themselves as a successful, occasionally even the best, approach to performing classification tasks. This usually involves supervised learning, where the developer of the ANN has decided in advance the classes and in which of these classes each observation should fit. (The less common usage with unsupervised learning, into which an ANN is used to cluster observations into classes, is not discussed in this paper). ANN have been particularly successful in classification tasks where the surfaces dividing classes are highly non-linear, and certainly out-perform linear classifiers for such applications.

Recently a number of algorithms for classification that require non-linear separating functions have been taken from the field of statistics, written in modern software and packaged with user-friendly development interfaces. Such software as AIM [AbTech, 1990] Modelware [Teranet IA, 1991] and MARS [Data Patterns, 1993] often perform as well or better than ANN and eliminate all the uncertainty inherent in the ANN training process. Detractors of such statistical tools for classification cite a loss of parallel processing as a weakness of such tools, and question the relative robustness of separator functions from statistics compared to those derived by of training an ANN.

The purpose of this paper is not to address the question of the relative robustness of ANN versus statistical classifiers but to present examples of two methods that use algorithms, one based on linear programming (LP), the other on machine induction, that as their output produce results which can be used to define immediately the architecture and weights of an ANN for the classification task. This is in stark contrast to the usual approach of training an ANN by trial and error, where for even a small classification task there may be perhaps 10 different architectures tried (primarily number of processing elements - neurons - in the hidden layer), 10 different parameters for the transfer function at each neuron and at least 10 runs with different starting points for each weight training process. Such an approach often takes literally days of computer time, with no guarantee of convergence, let alone a guarantee of finally identifying the optimal ANN architecture and weight matrix for the classification task at hand.

The next section defines notation for the classification problem and then explains how highly non-linear separation surfaces can be computed using LP. The section also explains how the output numbers from such a computation are used to define an architecture and weights for the classifying ANN.

Section 3 summarizes one of the many inductive approaches to classification and shows how the numeric outputs of this algorithm can be used to define an ANN for the classification. In Section 4 a small data set is used to illustrate the methods for defining an ANN by LP and machine induction and the Section also reports on the effort necessary to define and train an ANN for classifying the same data when back-propagation is used.

2. A Classifying ANN Defined by Linear Programming

The goal as defined here is to find parameters of functions that give a perfect separation between classes in the training set. These parameters are then used to define an ANN for the classification task. (The question of how the parameters may be modified to achieve a classifier that would be more "robust" in classifying observations not in the training set is beyond the scope of this paper).

We start the section by defining notation, beginning with the observations in the training set. An observation has n + 1 parameters, where n is the number of factors (or attributes) that describe the object, and the remaining parameter defines which class it is in. There are k = 1, .., K distinct classes. Considering just one class of observations, then suppressing any subscripting for the class, let there be m observations in the class. Then for a class define the matrix of observations A. Thus for some class k there is a matrix of observations

$$A = \begin{Bmatrix} a_{11} & \cdots & a_{1j} & \cdots & a_{1n} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ a_{i1} & \cdots & a_{ij} & \cdots & a_{in} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ a_{m1} & \cdots & a_{mj} & \cdots & a_{mn} \end{Bmatrix} \qquad (1)$$

Now consider an n-dimensional space where a point z has coordinates $\{z_j, j = 1, .., n\}$. A general quadratic expression in this space can be defined

$$\begin{aligned} &v_{11}z_1z_1 + v_{12}z_1z_2 \ldots\ldots + v_{1j}z_1z_j \ldots\ldots \quad v_{1n}z_1z_n \\ &\qquad + v_{22}z_2z_2 \ldots\ldots \quad + v_{2j}z_2z_j \ldots\ldots \\ &\qquad\qquad + v_{33}z_3z_3 \quad + \ldots\ldots \\ &\qquad\qquad\qquad\qquad + v_{nn}z_nz_n \\ &+ w_1z_1 \qquad + w_2z_2 \;+ \ldots\ldots \quad + w_jz_j \;+ \ldots\ldots + w_nz_n \\ &\qquad\qquad\qquad\qquad\qquad\qquad + x \end{aligned}$$

here $v_{jj'}$, j, j' = 1, ..., n represent the coefficients of the cross-products, w_j, j = 1, ..., n represent the coefficients of the linear terms and x is some constant.

Now let the vector z represent the values $\{a_{ij}, j = 1, ..., n\}$ of the factor observations for some object i (of class k) in the training set. Then the general quadratic expression is

$$\begin{aligned} &v_{11}a_{i1}a_{i1} + v_{12}a_{i1}a_{i2} \ldots\ldots + v_{1j}a_{i1}a_{ij} \ldots\ldots \quad v_{1n}a_{i1}a_{in} \\ &\qquad + v_{22}a_{i2}a_{i2} \ldots\ldots + v_{2j}a_{i2}a_{ij} \quad \ldots\ldots \\ &\qquad + v_{33}a_{i3}a_{i3} \quad + \ldots\ldots \\ &\qquad\qquad\qquad\qquad + v_{nn}a_{in}a_{in} \\ &+ w_1a_{i1} \quad + w_2a_{i2} \qquad + \ldots\ldots + w_ja_{ij} \qquad + \ldots\ldots + w_na_{in} \\ &\qquad\qquad\qquad\qquad\qquad\qquad + x = p_i \qquad (2) \end{aligned}$$

We are now ready to describe the LP procedure. The process presented here is the one defined by Ignizio and Baek [1992, 1993]. It is computationally more parsimonious than the approach described by Roy and Mukhopadhyay [1991]. Both seem to have been inspired by the work of Bennett and Mangasarian [1992] which in turn derived originally from

Mangasarian [1965, 1992]. The LP optimization is to determine those values of variables v_{ij} and w_j, j = 1 ... n and x in order to

Minimize $P = p_1 + p_2 .... + p_i .... + p_m$
subject to

$$v_{11} = -1 \tag{3}$$

and $p_i \geq 0$ i = 1 .... m, where p_i is defined above.

This LP describes in n-space a minimum quadratic surface (a spheroid, ellipsoid or paraboloid) that includes all the points in the class. This is referred to as a *supermask*. If it happens that this supermask includes some points that are from one or more of the other classes, these need to be eliminated. The procedure to achieve this is to find a *foreign* mask (or masks) that includes these foreign objects but none of the objects native to the class. More than one foreign mask may be required because of the limitations of the shape of a quadratic surface. This is illustrated in two dimensions in Figure 1.

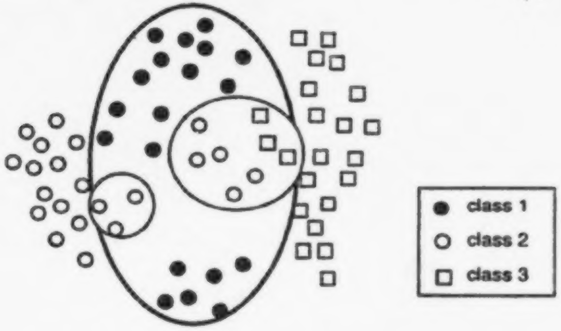


FIGURE 1: A Supermask and Foreign Masks

Once the supermask, and foreign masks, if they are required, have been identified according to the LP optimization, the final step is to turn the outputs from the LP into a specification of an ANN. There will be an input (distributor) node for each term in the quadratic expression that has a non-zero coefficient for at least one mask definition. In other words, if the observations have n factors, there will be up to n*n cross-products, up to n linear terms and one constant (unit) signal, for a total of up to $n^2 + n + 1$ input nodes. (The ANN computer chip may also contain circuits for computing the cross-products, but these are considered not to be part of the ANN). There will be one output neuron per class, that is K neurons, and neuron k will output a 1 if the inputs put the object in class k, and a 0 otherwise.

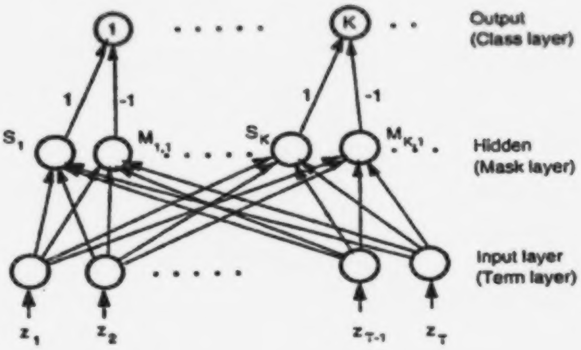


FIGURE 2: Neural Network Architecture

The ANN architecture is illustrated in Figure 2. In the hidden layer there is one neuron for each mask. The transfer function at each neuron of the hidden layer is a straightforward threshold function. A supermask neuron outputs a 1 if $\Sigma w_i x_i$ exceeds 0, otherwise it outputs 0. A foreign mask outputs -1 if $\Sigma w_i x_i$ exceeds 0_1 otherwise it outputs 0. Each neuron in the output layer simply outputs the (unit-weighted) sum of its inputs.

3. A Classifying ANN Defined from an Induction Algorithm

This section summarizes the results reported in an article by Goodman and Smyth [1993] that illustrates a method they had described in [Goodman and Smyth,1988 and Smyth and Goodman,1992].

A machine-learning induction algorithm develops a classification tree based on the value of one or more of the attribute values. For example, if attribute 1 has possible values 0 and 1, and the possible classes are A and B, then perhaps when attribute value 1 has value 0, most observations fall in class A, while if the same attribute has value 1, most of the observations in the training set fall in class B. In practice it is unlikely that a single attribute alone will be a sufficiently good classifier, so after a first "cut" with one attribute, the induction algorithm brings in a second attribute that "cleans up" the classification to give an output such as follows:

IF Attribute 1 = 0 ***and*** Attribute 2 = .5
THEN Class = A
with probability .91

The probability measure indicates a classification error of 1 - .91 = .09. The classification may be improved (smaller error) at the expense of bringing in yet another attribute and repeating until all attributes have been considered, or a perfect classification has been

TABLE I: Portion of the mutual funds raw database

Fund type	Diversity	Beta (risk)	Expense ratio (%)	Turnover rate (%)	Assets ($M)	5-year return
Growth	C	0.8	0.79	34	414.6	135.6
Growth	C	1.05	1.4	200	16.2	32.5
Gth&Income	A	0.96	1.34	127	26.8	88.3
AggrGth	A	1.23	1.4	161	64.2	24.4
Gth&Income	E	0.59	1.09	31	112.5	172.2
Gth&Income	A	0.58	1.18	64	76.4	80.4
Growth	B	0.86	1.6	54	11.7	52.7
Gth&Income	C	0.71	0.98	239	190	143.8
AggrGth	B	1.24	1.28	218	96.2	91.7
Growth	A	0.99	0.75	20	253.8	105.4
Growth	C	0.91	3.5	8	3.9	93

TABLE II: Categorized database as input to ITRULE

Fund type	Diversity <100% >100%	Beta	Stocks	Expns ratio (%) <1.5% >1.5%	Turn-over >100% <100%	Assets <$100M >$100M	Yield	Cap. gain	5-year return
Growth	Low	Under 1	Over 75%	Low	Low	Large	Under 3%	10-20%	Below S&P
Growth	Low	Over 1	Over 75%	Low	High	Small	Under 3%	Under 10%	Below S&P
Gth& Income	High	Under 1	Over 75%	Low	High	Small	Under 3%	10-20%	Below S&P
AggrGth	High	Over 1	Over 75%	Low	High	Small	Under 3%	Under 10%	Below S&P
Gth& Income	Low	Under 1	Under 75%	Low	Low	Large	Over 3%	Over 20%	Above S&P
Gth& Income	High	Under 1	Over 75%	Low	Low	Small	Over 3%	Under 10%	Below S&P
Growth	High	Under 1	Over 75%	High	Low	Small	Under 3%	10-20%	Below S&P
Gth& Income	Low	Under 1	Under 75%	Low	High	Large	Over 3%	Over 20%	Above S&P
AggrGth	High	Over 1	Over 75%	Low	High	Small	Under 3%	10-20%	Below S&P
Growth	High	Under 1	Over 75%	Low	Low	Large	Over 3%	Over 20%	Below S&P
Growth	Low	Under 1	Over 75%	High	Low	Small	Under 3%	Over 20%	Below S&P

achieved. In most practical applications a perfect classification of the training set will not be achieved, even if all attributes are considered, and usually a cut-off is specified such as achieving a 95% probability of correct classification or using only a limited number of the attributes.

If an attribute is not categorical but is on a continuous scale, it has to be discretized in some way. Goodman and Smyth's algorithm requires the developer to split the continuous scale into discrete ranges. Referring to the sample data of Table 1 for the characteristics and performance of a number of mutual funds, the non-categorical attributes are transformed by the developer to categorical variables as in Table II. Even the ordinal scale for diversity has been aggregated further to give a gross dichotomy Low-High. The data in Table II will be used for the LP and back-propagation demonstrations reported in the next section. The goal is to use attributes 1 - 9 to classify the mutual funds according to their 5-year return and whether this is above or below the Standard and Poor average of 138.5%

Goodman and Smyth's algorithm, named ITRULE, processes a set of observations to produce a number of rules of the form shown above using a single attribute or the conjunction of two or more attributes. As well as a probability measure ITRULE computes a measure of the utility of potential rules. The utility is an information theoretic measure defined

$$J[x;y=y]=p[y]\sum_x p[x|y]\log\left[\frac{p[x|y]}{p[x]}\right] \qquad (4)$$

This serves to measure the relative discernment of different rules, and the higher the utility measure, the more valuable the rule toward providing a classification.

In practice it is very unlikely that any single rule will give a perfect classification, that is with a probability measure of 1, so the algorithm combines rules with high utility, rather like combining opinions from different experts.

How this combination can be performed in an ANN is indicated in Figure 3. The input edge has input elements for every categorical value of every attribute to be used in attempting the classification. If an attribute has a value in a particular category, a signal of 1 is input at the node for that particular category, otherwise the input signal is 0. The first processing layer is made up strictly of inclusive AND gates, so the output from each of these neurons will be 0 or 1.

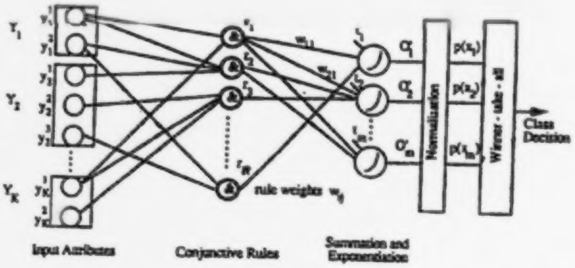


FIGURE 3: Neural Network Architecture form ITRULE

The weighting shown as w_{ij} in Figure 3 may be negative or positive, and is calculated by

$$w_{ij} = \log(p(x_i|LHS_j))/\log(p_i) \qquad (5)$$

where $p(x_i|LHS_j)$ is the probability value that has been calculated for the rule and p_i is a prior probability that the object observed is in category i. p_i is simply the relative frequency in the training set. Each neuron in the second processing layer also has a bias $t_i = \log p_i$. Thus if the rules provide no evidence for or against that class value, it retains its prior probability. The transfer function at the neurons receiving the weighted signals is first a standard summation $\Sigma_{j=1}^{r} w_{ij} r_j + t_i$. This is followed by an exponentiation (to convert from the logarithm back to the probability) and then a normalization to assure that the probabilities sum to one. Lastly the class with the highest probability is output as the prediction from that set of input values.

The resulting weights for the mutual fund sample data is illustrated in Figure 4.

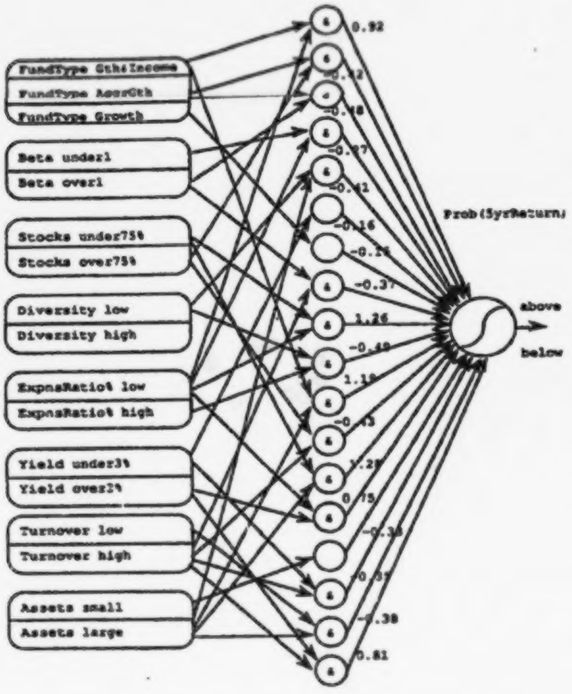

FundType Gth&Income
FundType AggrGth
FundType Growth
Beta under1
Beta over1
Stocks under75%
Stocks over75%
Diversity low
Diversity high
ExpnsRatio% low
ExpnsRatio% high
Yield under3%
Yield over2%
Turnover low
Turnover high
Assets small
Assets large
0.92
-0.12
-0.18
-0.27
-0.41
-0.16
-0.1
-0.37
1.26
-0.40
1.19
-0.43
1.28
0.75
-0.36
0.37
-0.38
0.81
Prob(5yrReturn)
above
below

FIGURE 4: Mutual Funds Neural Network

4. A Comparison of Computation by LP, Induction and Back-propagation

As an indication of the different computational effort to solve a classification problem by the three different methods, this section reports on classifying the mutual fund data listed in the previous Section using LP and back-propagation. Obviously the listed data set is too small to offer a comprehensive comparison, but is sufficient to illustrate the three approaches.

The LP was solved using the Optimizer tool in QuattroPro for Windows [Borland, 1992]. Using a microcomputer with a 386 chip and math coprocessor, the time between pressing the "Solve" command and the final answer appearing on the screen was less than one minute. No foreign masks were required for this data set, so the architecture and weights for the classifying ANN are determined immediately.

The back-propagation ANN learning was performed on a similar 386-based computer with math coprocessor as that running QuattroPro. The software used was MATLAB [Math Works, 1993] with the Neural Network Toolbox [Demuth and Beale, 1992]. An architecture with just one hidden layer was guessed to be sufficient and training was tried with successively 7, 5, 3, 2 and 1 nodes in the hidden layer. From these preliminary experiments it was judged that just 2 neurons in the hidden layer would be best to give fairly fast and stable convergence. A sum-of-squares error measure was used with an error goal of .01 and a limit of 1000 training epochs (repeated presentations of the complete training data set). The learning rate was set at 1.

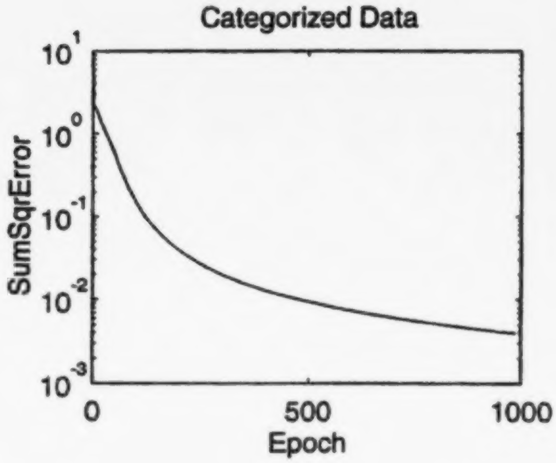


FIGURE 5: Back-propagation Training with Categorized Data

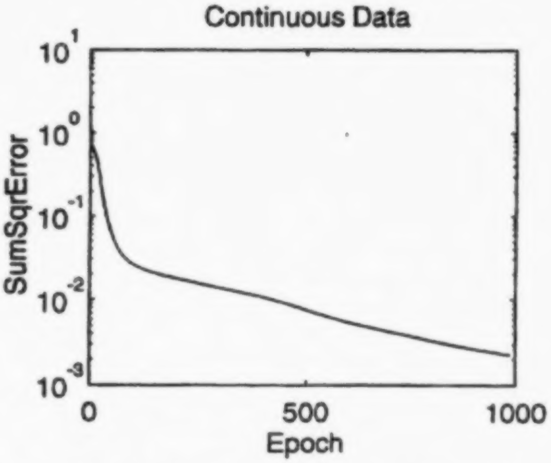


FIGURE 6: Back-propagation Training with Continuous Data

Training was attempted with both the categorized data of Table II and the original continuous data of Table I (after normalizing the continuous variables to a scale of 0 to 1). Convergence is illustrated by Figures 5 and 6. With the categorized data (Figure 5) convergence was slow, with over 100 epochs necessary to get down to even 10% sum of squares error and at 1000 iterations the error goal of .01 was not achieved. Convergence with the continuous data of Table I was faster though, even with this very simple and small data set, nearly 900 iterations were necessary to achieve the error goal. Training time for both sets of data was just under four minutes.

5. Conclusion

The computations shown here are simply for illustration, but they are a sufficient reminder of the effort usually necessary to decide the best architecture of an ANN and train it for classification by back-propagation. Any method that can give similar definition of ANN architecture and weights by computational methods that are both orders of magnitude faster and guaranteed to give correct answers is manifestly superior.

While this paper has not addressed the important issue of robustness of the finally designed ANN, it should be remembered that robustness in classification by an ANN is never guaranteed when training is performed by some ANN learning algorithm such as back-propagation. In fact achieving robustness usually requires another level of effort that once more requires trial and error and may increase the total computation time ten-fold, not to mention a comparable commitment of the precious time of the analyst. It is much more efficient to start with known optimal results from non-ANN algorithms and systematically introduce minor adjustments into the results of such algorithms as LP to obtain quickly the definition of a robust ANN for classification.

There is little similarity between the architecture and weights of the three ANN defined by the different methods of induction, LP and back-propagation. However, lest this be taken in some way to detract from deriving the ANN architecture and weights by LP or induction, it is worth remembering that the whole trial-and-error approach used in deciding an acceptable architecture and weights for an ANN trained by back-propagation (or any other ANN learning procedure) is heuristic, and there is no guarantee that the best ANN architecture and weights will be found for the classification task at hand. There is also no way other than experimentation to determine the significance of the weights, and thereby estimate some measure of robustness. In contrast, the architecture and weights obtained by the LP or induction approach have immediate significance, which can serve as a basis for estimation the robustness of the ANN in dealing with data outside the training or the test data sets.

References

AbTech Corporation, [1990] *AIM Documentation*, AbTech Corporation, Charlottesville, VA, 1990.

Bennett, K.P. and Mangasarian, O.L., [1992] "Neural Network Training via Linear Programming," in P.M. Pardalos (ed.) *Advances in Optimization and Parallel Computing*, North Holland, Amsterdam, pp. 55-67, 1992.

Borland Inc., [1992] *QuattroPro for Windows: User's Guide*, Borland International Inc. Scotts Valley, LA. 1992.

Data Patterns, [1993] *MARS Documentation*, Data Patterns, Philadelphia, PA, 1993.

Dantzig, G.B., [1948] "Programming in a Linear Structure," Comptroller, USAF, Washington D.C. February 1948.

Demuth, H. and Beale, M., [1993] *The Neural Network Toolbox for MATLAB: User's Guide*, The Math Works Matick, MA 1992.

Goodman, R.M. and Smyth, P., [1989] "The induction of probabilistic rule sets - the ITRULE algorithm", *Proceedings of the Sixth International Workshop on Machine Learning*, Los Altos, CA: Morgan Kaufman pp 129-132.

Goodman, R.M. and Smyth, P., [1993] "Automated Induction of Rule-based Neural Networks from Databases," *Intelligent Systems in Accounting, Finance and Management*, vol. 2, no. 1, January 1993.

Ignizio J.P. and Baek, W., [1992] "An Alternative Neural Network Architecture and Training Algorithm," Working Paper, Department of Systems Engineering, University of Virginia, July 1992.

Ignizio, J. and Baek, W., [1993] "Knowledge Programming Meets Linear Programming II," *PC AI*, vol. 7, no. 1, pp. 45-49 January/February 1993.

Mangasarian, O.L., [1965] "Linear and Nonlinear Separation of Patterns by Linear Programming", *Operations Research*, vol. 13, pp. 444-452, 1965.

Mangasarian, O. L., [1992] "Mathematical Programming in Neural Networks," Computer Sciences Technical Report #1129, University of Wisconsin-Madison, December 1992.

Math Works, [1993] *MATLAB User's Guide,* The Math Works Inc. Matick, MA, 1993.

Roy, A. and Mukhopadhyay, S., [1991] "Pattern Classification Using Linear Programming," *ORSA Journal on Computing,* vol. 3 no. 1, pp. 66-80, Winter 1991.

Smyth, P. and Goodman, R.M., [1992] "An Information Theoratic Approach to Rule Induction from Databases" *IEEE Transactions on Knowledge and Data Engineering,* Vol. 4, No. 4, pp. 301-316, August 1992.

Teranet IA, [1992] *ModelWare User's Manual* Teranet IA, Incorporated, Nanaimo, B.C. 1992.

905C20

Face Identification Based on an Interactional Neural Network

Pierre Boulanger[1]
Institute for Information Technology
National Research Council of Canada
Ottawa, Ontario
K1A 0R6

#149756

Abstract

The problem of the computation of weights in neural networks performing arbitrary mapping from input to output is solved with assumption of a new discriminatory condition. The algorithm producing the weight matrix is implemented in the form of a self-programming neural network. Such an implementation requires a layer of hidden neuromorphic processors (dipoles). The two-layer network (interactional neural network) and the process of self-programming are presented.

The self-programming of the interactional network and its ability to recognized images are illustrated by a computer simulation of memorized images of different human faces. The system first scan the face using a range sensor from which one computes the position of depth and orientation discontinuities. The discontinuity image is then used to train the network in the learning mode or to stimulate the network to identify a particular face in the identification mode.

The paper present experimental results illustrating the capability of the network to discriminate between faces and its insensitivity to noise.

I. Introduction

One of the major focus of the neural network research are learning algorithms. The ability to learn arbitrary mapping from input to output allows neural networks to be considered as a content-addressable memory [1] for the control of behaviour. The memory is trained to match specific input stimuli and desired output "reactions" for applications in robotics [2], [3], [4], [5].

The learning procedures consist of correction of output errors through connection strength modulation (supervised learning). The increment or decrement of a weight (strength of a connection in the network) depends functionally on output deviations in a preceding step and actual training input. Typical learning is the procedure which ensues the *delta rule*[6] formula: the increments $\Delta w(i,j)$ (of weights from the input units i to the output units j) are proportional to training input in i times output error in j. So, locally available information (in i and in j) determines $\Delta w(i,j)$ (the principle of Hebbian learning).

In the model developed in this paper, internal representations (*cf.* [11], and [12], ch. 8) of input stimuli are created and a recognition of the input precedes any action of associating input "images" and output "reactions". In this way the function of representing and recognizing is split off from the input-to-output mapping. The ability to recognize specific individual inputs $\mathbf{I}_1, \mathbf{I}_2, \ldots$ among all possible stimuli $\mathbf{I}$ is *programmable* in the sense of parallel distributed processing [12]: the local training input $I_k(i)$ adjusts the weights of connections with i in order to "memorize" $\mathbf{I}_k$ without disrupting other previously learned inputs.

Let us give here some mathematical precision for the introductory notions. The content-addressable memory consists of parallel processing of input patterns which are distributions of activations on *input units*. These are enumerated as *input neurons* $1, 2, \ldots, N$. The input neurons are allowed to take on one of the two *activation values* 1 and 0, i.e., the input patterns are binary vectors $\mathbf{I} = [I(1), I(2), \ldots, I(N)]$, $I(i) = 0$ or 1.

In a *single layer neural network* a neuron i is connected to *output units* O_j, $j = 1, \ldots, M$. The *weights* $w(i,j)$ of respective connections is represented by a $N \times M$ *weight matrix* $\mathcal{W}$. So, a unit O_j collecting the *net input* is expressed by

$$\sum_{i=1}^{N} I(i)w(i,j) = net_{\mathbf{I}}(j).$$

Abstracting any threshold property of output units, one can define the linear mapping

$$\mathbf{I} \rightarrow \mathbf{net_I} = [net_{\mathbf{I}}(1), net_{\mathbf{I}}(2), \ldots, net_{\mathbf{I}}(M)] = \mathbf{I}\mathcal{W}$$

to be the function of the network. Two networks are *linearly equivalent* if they perform the same function.

The threshold 0 of activation for the output units causes a bipolar, i.e., 1 or -1, *output activation pattern* $\mathbf{A_I}$ ("reaction") to appear on the output layer.

A learning procedure evolves weights $w(i,j)$ to produce the matching of proper "reactions" $\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_K$ to specific stimulus $\mathbf{I}_1, \mathbf{I}_2, \ldots, \mathbf{I}_K$. A typical learning algorithm consists in updating of the weights of connections until the sum of the squared error for input/output expressed by

---

[1] NRC 37052

$$E = \frac{1}{2}\sum_{k=1}^{K} \|\mathbf{A}_k - \mathbf{A}_{\mathbf{I}_k}\|^2,$$

attains a minimum (which can be local, *cf.* [12], ch. 8).

The process of self-programming presented here will find a weight matrix fulfilling the following requirement

$$\mathbf{A}_{\mathbf{I}_k} = \mathbf{A}_k, \;\; k = 1, 2, ..., K, \;\; \text{i.e. } E = 0.$$

In fact the matrix will be a solution of the linear problem

$$\mathbf{I}_k * \mathcal{W} = \alpha_k \mathbf{A}_k, \;\; k = 1, 2, ..., K \qquad (1)$$

where α_k are positive coefficients. So, each $\mathbf{net}_{\mathbf{I}_k}$ will be a scalar copy of $\mathbf{A}_k$. A sufficient requirement for the algorithm to work is the *discriminatory condition* for inputs $\mathbf{I}_1$, ..., $\mathbf{I}_k$. The condition is stronger than the linear mutual independence in the set of input vectors (at most $N-1$ vectors can be discriminatory in the space of dimension N).

To solve the problem we consider the pairs of input neurons as parallel processors. We will call such processor *interactions*. The interaction are a mathematical notion, but can be physically embodied in a layer of hidden neurons [1]. The three–layer network called by Porada and Boulanger [14] an *interactional neural network*, where hidden neurons, each connected to two inputs, transmit the difference of the two inputs to the output units. One will demonstrate in this paper that this new method of training associative memory is very attractive.

In Section II, we will describe the mathematical foundation of the problem to be solved. In Section III the interactional model will be presented and in Section IV the process of self-programming is described. Section V present a computer simulation of this algorithm and experimental results on the recognition of faces.

II. Memory Matrix Problem

Let us write the system of equations (1) in matrix form. Denoted by $\mathcal{I}$ the $K \times N$ matrix where the row vectors are the input vectors $\mathbf{I}_1$, ..., $\mathbf{I}_K$ (written top-down). Analogously, the vectors $\mathbf{A}_1$, ..., $\mathbf{A}_K$ are columns of a $K \times N$ matrix $\mathcal{A}$. Now, (1) is equivalent to

$$\mathcal{I}\mathcal{W} = \mathcal{D}\mathcal{A} \qquad (2)$$

where $\mathcal{D}$ is a diagonal $K \times K$ matrix and the diagonal terms are positive.

Suppose that one have constructed a $N \times K$ matrix $\mathcal{V} = [v(i,k)]$ such that

$$\mathcal{I}\mathcal{V} = \mathcal{D} \qquad (3)$$

In this case, the weight matrix

$$\mathcal{W} = \mathcal{V}\mathcal{A}$$

is a solution of (2). So, it is sufficient to find $\mathcal{V}$ verifying (3), the factor $\mathcal{A}$ being a facultative matrix of desired "reactions".

If the weight matrix of a neural network is a product (actually $\mathcal{W} = \mathcal{V}\mathcal{A}$), then the network itself can be realized as an arrangement of two networks in cascade: the first one is equivalent to the weight matrix $\mathcal{V}$ and its net output becomes the input to the second one, equivalent to the memory matrix $\mathcal{A}$ (*cf.* [13], and [12]). That system is a *two- layer* network. An *intermediate unit* I_k ($k = 1, ..., K$) is activated by the aggregate signal

$$\sum_{i=1}^{N} I(i)v(i,k) = \alpha_{\mathbf{I}}(k)$$

which is weighted by $A_k(j)$ and send forth to O_j, $j = 1, 2, ..., M$. It can be easily shown that

$$\mathbf{net}_{\mathbf{I}} = \alpha_{\mathbf{I}}(1)\mathbf{A}_1 + ... + \alpha_{\mathbf{I}}(K)\mathbf{A}_K. \qquad (4)$$

The first network, when stimulated by $\mathbf{I}$, furnishes the coefficients $\alpha_{\mathbf{I}}(k)$. Its weight matrix $\mathcal{V}$ is a solution of (3) if, and only if,

$$\alpha_{\mathbf{I}_k}(k) > 0, \quad \alpha_{\mathbf{I}_k}(l) = 0 \text{ for } l \neq k \qquad (5)$$

i.e., the matrix $[\alpha_{\mathbf{I}_k}(l)]$ is diagonal.

Coming back to the activation rule for O_j, we see that an input $\mathbf{I}$ will be associated to $\mathbf{A}_k$ if, for example,

$$\sum_{l \neq k} |\alpha_{\mathbf{I}}(l)| < \alpha_{\mathbf{I}}(k) \qquad (6)$$

(under that condition, by (4), $\text{sign}(net_{\mathbf{I}}(j)) = A_k(j)$).

The only task of the second layer of connections is to calculate the linear combination (4) of optional reactions using the output $[\alpha_{\mathbf{I}}(1), ..., \alpha_{\mathbf{I}}(K)]$ from the first layer. This procedure and the second layer itself will be omitted in further considerations. So, the intermediate units I_k will stand for output units and $\alpha_{\mathbf{I}}$ will be the output activations. The condition (5) is now the test for $\mathcal{V}$ to be the solution of the weight matrix problem (3).

III. Interactional Model

The mathematical problem of a matrix $\mathcal{V}$ fulfilling (3) can be solved by means of a neural network with *hidden* units. A detailed analysis of such systems and their discriminatory capabilities is provided in [11], see also [12], ch. 8. We consider a model, called *interactional neural network* [14] where a hidden neuron (*dipole*) connects with *two* input neurons, with one excitatory and one inhibitory connection (see [12], ch. 5, "Dipole Experiments"). Let us see how such networks are related to our problem.

The interactional network will be a device carrying out a linear mapping

$$\mathbf{I} \rightarrow \alpha_{\mathbf{I}} = [\alpha_{\mathbf{I}}(1), ..., \alpha_{\mathbf{I}}(K)] \qquad (7)$$

which depends on the collection of dipoles involved in the network and on the weights of connections between the dipoles and output units. The dipoles responsible for discrimination of the inputs $\mathbf{I}_1, \dots, \mathbf{I}_K$ must be revealed. The weights which guarantee (5) will be given in explicit form. But the linearity of the interactional network and a "duality theorem" described in [14], lead to a "classical" single layer network linearly equivalent to the interactional one. The weights of connections in the single layer network will be, finally, a solution of (3).

Some *ordered pairs* (i', i'') of input neurons are considered as dipole processors embodied in hidden neurons. The dipole is *directed*, i.e., the function of its two entries is not symmetrical. Thus, the hidden neuron h receives the input $I(i')$ from i' and $I(i'')$ from i'' and outputs the difference $I(i') - I(i'')$. Such an activation rule for h stands for the assumption that the connection between the second input neuron and h is inhibitory. We consider only the networks where one unordered pair i', i'' involves at most one dipole (corresponding to (i', i'') or to (i'', i')).

A particular interactional network can be described in terms of a *directed graph* Γ over the layer of input neurons (which become vertices of the graph). Each directed edge $e = (i', i'')$ of the graph corresponds to a dipole $h = h_e$ which sends forth the signal

$$I^*(h_e) = I(i') - I(i'') = 1,\ 0 \text{ or } -1$$

when a stimulus $\mathbf{I}$ activates the input neurons. The vector $\mathbf{I}^* = [I^*(h_e)]_{e \in \Gamma}$ is the *internal representation* of the input $\mathbf{I}$.

Consider, for a moment, arbitrary weights $v^*(h_e, l)$ for the connections between h_e and I_l. A unit I_l is activated by

$$\sum_{e \in \Gamma} I^*(h_e) v^*(h_e, l)$$

and so the network performs a stimulus-to-output activations mapping which is the matrix operator

$$\mathbf{I} \longrightarrow \mathbf{I}^* \mathcal{V}^*. \tag{8}$$

There is a single layer network functionally equivalent to the interactional one. Its weight matrix can be produced as follows:

Denote, for a vertex i of Γ, by L_i the collection of edges *leaving* i (so, i is the *first* vertex of the directed edges), and by E_i the collection of edges *entering* i (i is the *second* vertex of edges in E_i). Let

$$v(i, l) = \sum_{e \in E_i} v^*(h_e, l) - \sum_{e \in L_i} v^*(h_e, l). \tag{9}$$

If an input unit i connects with I_l with the weight $v(i, l)$ then the network performs the function (8).

This statement is equivalent to the "duality theorem"

$$\mathbf{I}\mathcal{V} = \mathbf{I}^* \mathcal{V}^* \quad \text{for any } \mathbf{I}$$

proved in [14]. Now the condition (3) becomes

$$\mathcal{I}^* \mathcal{V}^* = \mathcal{D} \tag{10}$$

where $\mathcal{I}^*$ is the matrix of internal representations of $\mathbf{I}_1, \dots, \mathbf{I}_K$.

IV. Self-programming Cycles

Triangular Self-programming

The self-programming algorithm is a process of autonomous creation of dipoles step by step, in K *cycles*, supplying successively $\mathbf{I}_1, \dots, \mathbf{I}_K$ as training inputs. At the same time the weights of connections between actually appearing dipoles and the output units are to be determined.

The expression "autonomous" means that a new edge (i', i'') (in fact the respective dipole) appears in the k^{th} cycle depending only on local information $I_k(i')$, $I_k(i'')$. Similarly, the new weights must be derived from the output of an actual network stimulated by a new training input.

A priori, any pair of input neurons may be needed as part of a hidden processor. All pairs i', i'' form a virtual space where, mathematically, the graph Γ is inscribed and, practically, the dipoles are attached. A convenient approach is the supposition of "virtual" dipoles: before any programming takes place, each pair i', i'' of input neurons is connected to one hidden neuron h. The two inputs of h being symmetrical, the neuron outputs the signal $|I(i') - I(i'')|$. The virtual neurons are not connected to the output units.

The interactional neural network will be inscribed in this memory by directing virtual neurons and setting weighted connections between them and $I_1, \dots, I_K$.

The vector $\mathbf{I}_1$ serves as *training input* in the first cycle of self-programming. It produces a 0,1 activation pattern in the layer of virtual neurons. The value 1 of activation causes the virtual neuron to be "switched on" and properly directed. So, if i' and i'' are the two inputs of h and $|\mathbf{I}_1(i') - \mathbf{I}_1(i'')| = 1$ then h becomes the dipole h_e. Precisely, if $I_1(i') > I_1(i'')$ then $e = (i', i'')$, and for any future stimulus $\mathbf{I}$ the neuron will output $I^*(h_e) = I_1(i') - I_1(i'')$. If $I_1(i') < I_1(i'')$ then $e = (i'', i')$ and $I^*(h_e) = I(i'') - I(i')$. Now h_e is connected to I_1 and the weight of connection is set to 1.

Coming back to the mathematical description of the interactional network, we see that the dipoles which appeared in the first cycle correspond to edges of index 1. The weights coincide with the first row of $\mathcal{F}$ (see (12), the weight 0 stands for absence of connection) and form the first package of identical rows of the matrix $\mathcal{V}^*$. At this moment the stimulus $\mathbf{I}_1$ produces the activation $I_1^{**}(1)$ for I_1 and 0 for other output units.

Let us describe the k^{th} cycle.

Actually, the population of dipoles corresponds to edges of an index $k' \leq k - 1$. The k'-th row $[f(k', 1), \dots, f(k', K)]$ of $\mathcal{F}$ represents the weights of connections between a dipole h_e, where $\kappa(e) = k'$, and $I_1, ..., I_K$. We have $f(k', l) = 0$ for $l \geq k'$. Other hidden neurons remain virtual and do not contribute to the output activations. So the actual network is linearly equivalent to the mathematical model, where $f(k'', l) = 0$ if $k'' \geq k$.

Let us stimulate such a network by $\mathbf{I}_k$. Only the units I_l, $l < k$, are activated and

$$I_l = \sum_{k'=1}^{k-1} \sum_{\kappa(e)=k'} I_k^*(e)\mu(h_e, l) = \sum_{k'=1}^{k-1} I_k^{**}(k') f(k', l)$$

The stimulus $\mathbf{I}_k$ will also "switch on" and direct those virtual neurons h where the actual activation value is 1. So, if h is attached to i' and i'' and $|I_k(i') - I_k(i'')| = 1$, than one new edge $e = (i', i'')$ or $e = (i'', i')$ is defined by the requirement $I_k^*(h_e) = 1$. The neuron h becomes the dipole h_e and obtains a connection with I_k, weighted by 1 where the activation of I_k is equal to

$$I_k = I_k^{**}(k)$$

The set of edges defined in this cycle corresponds to the edges with index k in the mathematical description of Γ.

The activations of $I_1, ..., I_k$ determine the weight $f(k, l)$ of connections between a dipole appearing in this cycle and I_l, $l < k$:

$$f(k, l) = -\frac{\text{activation of } I_l}{\text{activation of } I_k}$$

We see that the new connections obtain the weights, those which were the solution of our weight matrix problem.

Diagonal Self-programming

The model of interactional neural network and its self-programming algorithm take a simpler form if the input vectors verify the following discriminatory condition:

Definition 1. For any k there exist pairs i', i'' of input neurons such that

$$I_k(i') \neq I_k(i'') = 1 \text{ and } I_l(i') = I_l(i'') \text{ for } l \neq k$$

In this case we can admit switching off the hidden units attached to "ambiguous" pairs i', i''. The pair is "ambiguous" if $I_k(i') \neq I_k(i'')$ for more than one k. A unit, once switched off, cannot be reactivated.

Thus, consider the *diagonal* self-programming of the initial memory. When $\mathbf{I}_k$ is the training input (k^{th} cycle), the reprogramming of a hidden unit h (attached to i' and i'') is managed by the following algorithm:

```
1. IF $I_k(i') - I_k(i'') \neq 0$ THEN
2. IF $h$ is still virtual THEN
3. IF $I_k(i') - I_k(i'') = 1$ THEN $h = h_e$ where $e = (i', i'')$
4. IF $I_k(i') - I_k(i'') = -1$ THEN $h = h_e$ where $e = (i'', i')$
5. ELSE $h$ is switched off and the corresponding directed edge is erased
6. ELSE $h$ passes with no change to the next cycle
```

Remark that Γ consists of directed edges which appeared in one cycle and survived the succeeding cycles. So, Γ is the set of all pairs $e = (i', i'')$ such that

$$I_k(i') \neq I_k(i'') \text{ for exactly one } k.$$

Output representations are now identical with output activations: $\alpha_{\mathbf{I}} = \mathbf{I}^{**}$, accordingly to the fact that the diagonal self-programming algorithm produces a matrix $\mathcal{F}$ which is simply the $K \times K$ unitary matrix.

V. Experimental Results

The number of all possible hidden neurons in a interactional network is $\frac{1}{2}N(N-1)$, where N is the number of input units. But an initial memory where all virtual neurons are present is a mathematical tool rather than a practical device. The discrimination and efficiency of self-programming is ensured mostly by the "short-range" edges of the interactional graph Γ. It is a matter of further mathematical examination, but all simulations we have made show that the best discrimination properties have "shapes", i.e. "images" defined as the interior of an outline. In these cases, the huge population of "long" edges is redundant, even if the number K of training images is comparable with N.

The interactional neural network and the self-programming were simulated on a SGI VGX machine. The input stimuli are binary vectors of length $N = 38400$ arranged in 2-D 150×256 arrays. So $I(p, q)$ ($1 \leq p \leq 149$ and $1 \leq q \leq 255$) stands for $I(i)$, where $i = 256(p-1) + q$.

One family of the training input vectors is shown in Figure 1. These $K = 6$ images are corresponding to depth and orientation discontinuities of human faces computed from range images extracted from NRCC Image Database [15] using a method described in [16].

Such a small family fulfils, with a lot of redundancy, the discriminatory condition in the sense of Definition 1, so, only the *short-range* interactions are taken into account. One input neuron corresponds to one couple (p, q) of array coordinates and for short-range interactions we consider only the pairs of planar points (p', q'), (p'', q'') such that

$$|p' - q'| \leq 2 \text{ and } |p'' - q''| \leq 2. \qquad (11)$$

The virtual neurons are stored into a data structure where each elements contains one pair of planar points (p', q'), (p'', q'') and a variable κ, set

Img	A_0	A_1	A_2	A_3	A_4	A_5
a	30567	0	0	0	0	0
b	824	55910	0	0	0	0
c	860	1042	22070	0	0	0
d	349	-25	30	24570	0	0
e	332	1466	1193	463	18788	0
f	779	441	166	-32	207	15955

Table 1: Response of the network to the stimulation of images illustrated in Figure 1.

initially to -1. The structure represents the initial memory. The variable κ will store $\kappa(e)$ if the record will become a dipole h_e. In this case the pair of planar points (p', q'), (p'', q'') is eventually rearranged to have $I_\kappa(p', q') - I_\kappa(p'', q'') = 1$.

The matrix $\mathcal{F} = [f(k, l)]$ (recall that $f(k, l)$ is the weight of connections between the dipoles where $\kappa = k$ and I_l) will also be created in the process of simulated self-programming. Thus, a $K \times K$ array $[f(k, l)]$ and a vector $\mathbf{I}^{**} = [I^{**}(1), ..., I^{**}(K)]$ are declared in the procedure. The 150×256 array $\mathbf{I} = [I(p, q)]$ will store one successive training input and $\mathbf{I}^{**}$ its output representation.

After the 6 cycles of self-programming, the structure of hidden neurons and the weight matrix $\mathcal{F}$ represent a "memory" which can be stimulated by any input array I_k. The results of the stimulation can be found in Table 1. The diagonal elements correspond to the identified image. It is obvious from these results that the network is highly discriminant.

The procedure of stimulation simulates the execution of the operator (7) by an interactional network. The initial values $\alpha_{\mathbf{I}}(l)$ are set to 0. The "memory" structure is read and if $\kappa > 0$ then the assignments

$$\begin{array}{c} \alpha_{\mathbf{I}}(1) \leftarrow \alpha_{\mathbf{I}}(1) + (I(p', q') - I(p'', q''))f(\kappa, 1) \\ \vdots \\ \alpha_{\mathbf{I}}(K) \leftarrow \alpha_{\mathbf{I}}(K) + (I(p', q') - I(p'', q''))f(\kappa, K) \end{array}$$

are executed. The final values of $\alpha_{\mathbf{I}}(1)$, ..., $\alpha_{\mathbf{I}}(K)$ are returned.

Stimulations by $\mathbf{I}_1$, ..., $\mathbf{I}_{12}$ accentuate only the mathematical result (10). Thus, the coefficients $\alpha_{\mathbf{I}_k}(l)$, $1 \leq k, 0 \leq 5$, of the matrix $\mathcal{I}^*\mathcal{V}^*$ form a diagonal matrix $\mathcal{D}$ with positive diagonal terms.

A procedure executing the diagonal self-programming can be found in appendix A. The "memory" is represented by a structure storing the dipoles h_e and their indices $\kappa(e)$. A dipole h_e connects only with $I_{\kappa(e)}$ (with weight 1), so the stimulation by an array $[I(p, q]$ consists in reading the file and assigning

$$\alpha_{\mathbf{I}}(\kappa) \leftarrow \alpha_{\mathbf{I}}(\kappa)(I(p', q') - I(p'', q''))$$

when $\kappa > 0$.

We have also tested the network for noise sensitivity. One can see in Figure 2 the image illustrated in Figure 1a contaminated by an increasing amount of noise. Table 2 illustrate the corresponding network response to the contaminated image. As one can see the network is quite insensitive to noise even at a level of 75%.

Img	A_0	A_1	A_2	A_3	A_4	A_5
a+10%	27737	436	-5	277	148	40
a+30%	22978	490	340	209	101	-103
a+50%	19102	633	393	58	203	89
a+75%	15504	1062	243	-92	137	64

Table 2: Response of the network to a stimulation of Figure 1a corrupted by noise as illustrated in Figure 2.

VI. Conclusions

When designing a neural network for the control of behaviour, the interactional model provides a solution for the problem of autonomous learning and recognition of specific stimuli. Thus, the self-programming is, in the first place, a method of predetermined learning: it furnishes the optimal memory matrices for the content addressable controller. But, using the dipole processors, we are able to implement the interactional model as the control unit itself. In this case the self-programming is a session of on-line learning.

The question is in what measure the discriminatory condition we have assumed delimitates the number of recognizable input stimuli. Our conjecture is that a set of input vectors can not be arranged in a discriminatory sequence in the sense of Definition 1 if, and only if, the vectors are independent in the sense of *affine* spaces. The affine independence means that the vectors *and* the vector $[1, 1, ..., 1]$ are mutually linearly independent. Another problem is the ordering of input training stimuli in regard to the triangular self-programming. For example, when learning three input vectors $[0, 1, 1, 0]$, $[0, 0, 1, 1]$, $[0, 1, 1, 1]$, the vector $[0, 1, 1, 1]$ can not be learned in the last cycle of the triangular self-programming and must occur in first or second cycle. An algorithm to perform this ordering is not known.

References

1. T. Kohonen, *Content-Addressable Memories*, Vol. 1. Springer- Verlag, Berlin, 1980.

2. A. G. Barto, R. S. Sutton, and Ch. W. Anderson, "Neuronlike Adaptive Elements That Can Solve Difficult Learning Control Problems", *IEEE Trans. Systems, Man, and Cybern.*, Vol. SMC-13, No. 5, 834-846, 1983.

3. A. Gucz, J. Selinsky, "A trainable Neuromorphic Controller", *Journal of Robotic Systems*, Vol. 5, No. 4, 363-388, 1988.

4. J. Tanomaru and S. Omatu, *Towards effective neuromorphic controllers*, Proceeding of IECON'91, Vol. 2 pp. 395–400, 1991.

5. N.G. Bourbakis and F. Barlos, *Hardware design of the lower level nodes of HERMES neuromorphic net*, Journal of Eng. Applications of Artificial Intelligence, pp. 23–31, 5(1), Jan. 1992.

6. R. S. Sutton, A. G. Barto, "Toward a modern theory of adaptive networks: Expectation and prediction", *Psychological Review*, 135-170, 1988.

7. J. J. Hopfield, "Neural Networks and Physical Systems with Emergent Collective Computational Abilities", *Proc. Natl. Acad. Sci. USA*, Vol. 79, 2554-2558, 1982.

8. G. A. Carpenter, S. Grossberg, "Neural Dynamics of Category Learning and recognition: Attention, Memory Consolidation, and Amnesia", in J. Davis, R. Newburgh, and E. Wegman (Eds.) *Brain Structure, Learning, and Memory*, AAAS Sumposium Series, 1986.

9. D. H. Ackley, G. E. Hinton, and T. J. Sejnowski, "A learning algorithm for Boltzmann machines", *Cognitive Science*, Vol. 9, 147-169, 1985.

10. P. Smolensky, "The mathematical role of self-consistency in parallel computation", *Proceedings of the Sixth Annual Conference of the Cognitive Science Society*, 1984.

11. M. Minsky and S. Pappert, *Perceptrons*, Cambridge, MA: MIT Press, 1969.

12. D. E. Rumelhart , J. L. McClelland (Eds.), *Parallel Distributed Processing*. Vol. 1. Cambridge: MIT Press. 1986.

13. J. L. McClelland, "On the time-relations of mental process: An examination of systems of processes in cascade", *Psychological Review*, Vol 86, 29-63, 1979.

14. E. Porada and P. Boulanger "Interactional Parallel Processing in Neural Nets", *Proceeding of the International Joint Conference on Neural Networks*, Washington, June 1989.

15. M. Rioux, L. Cournoyer, *The NRCC Three-dimentional Image Data Files*, Ottawa: NRCC Report 29077, 1988.

16. P. Boulanger, F. Blais, and P. Cohen, *Detection of depth and orientation discontinuities in range images using mathematical morphology*, 10th International Conference on Pattern Recognition, Atlantic City, USA, June 1990.

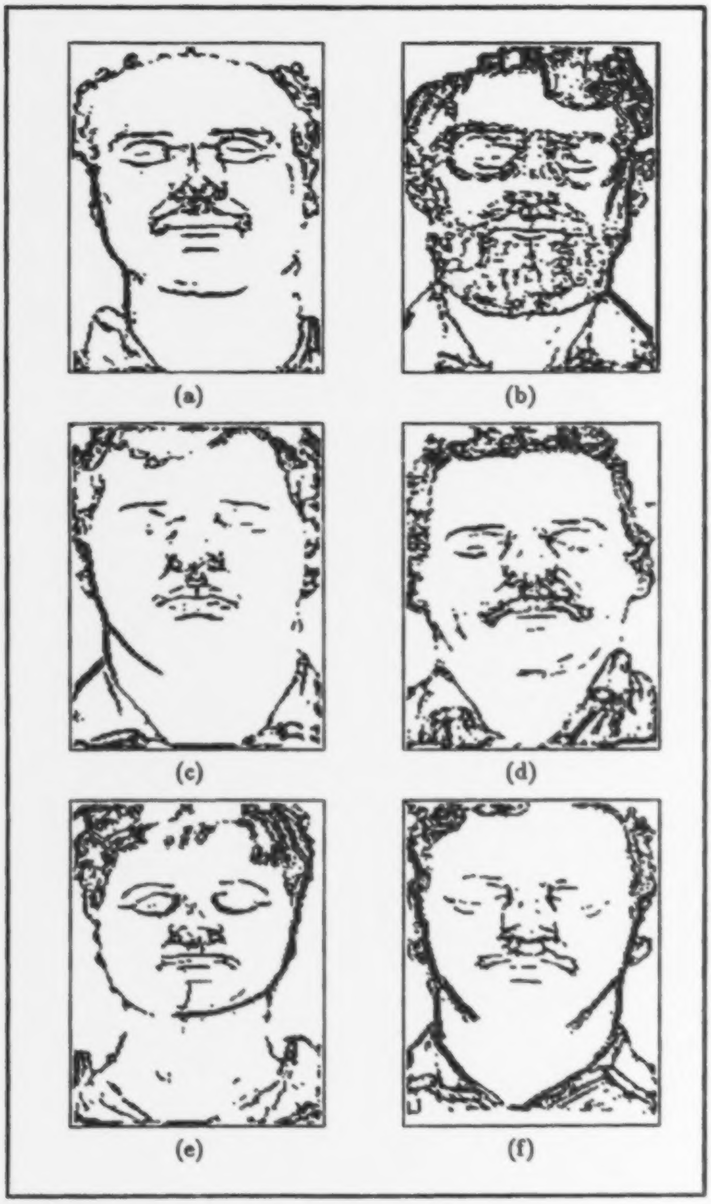
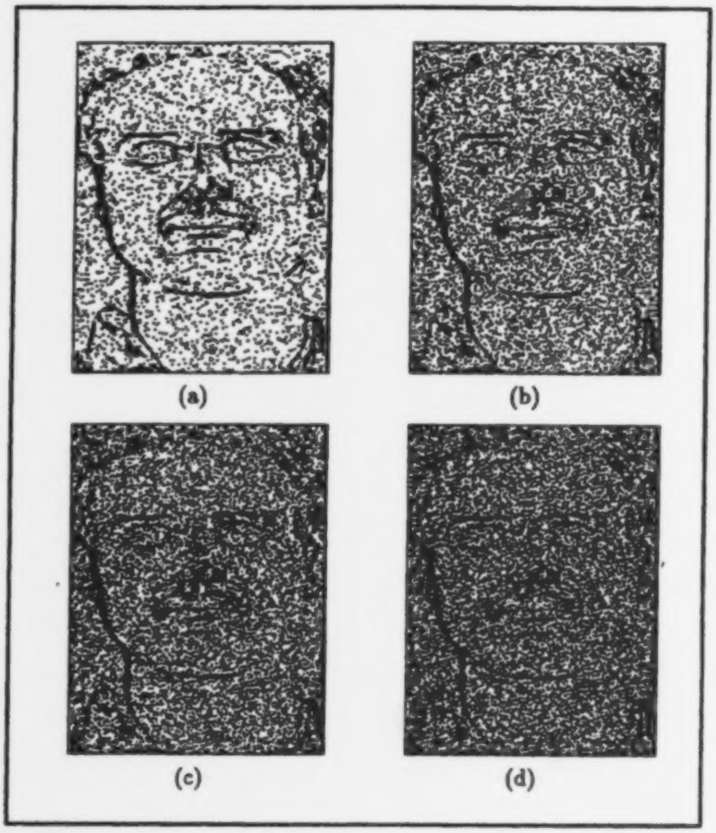


Figure 1: Depth and orientation discontinuities of range images of multiple faces.



(a) (b) (c) (d)

Figure 2: Image of a face corrupted by an increasing amount of noise: (a) 10%; (b) 30%, (c)50%, and (d) 75%

90C201

Neural Network and Knowledge-Based Expert Systems for Signal Validation in the Nuclear Industry

Heping Dai and J. Walter Thompson
Atlantic Nuclear Services Ltd., P. O. Box 1268, Stn. A.
500 Beaverbrook Ct., Fredericton, N.B. Canada E3B 5C8

#149757

Abstract

A knowledge-based expert system for signal validation in the nuclear industry, incorporating the results of neural network modelling, is described in this paper. In particular, the pressure and inventory control system of a CANDU reactor has been investigated. Several neural network models have been developed and used to predict signals of interest during signal validation. Predictions of key variables in the process have a high degree of accuracy, even during transient operations such as fuelling, and various types of system testing. Frames are used to represent knowledge in constructing the knowledge-based expert system for signal validation. The entire expert system is developed in both C and Prolog. From the practical point of view, issues about potential applications of the proposed system in industry are discussed.

Introduction

Recently, extensive attention has been focused on the validation of signals generated from complex physical processes in industry. In particular, the importance of signal validation in the nuclear industry has become more and more clear. The CANDU (CANadian Deuterium Uranium) reactor is a typical example, where control of the reactor requires fast and reliable processing of large amounts of data sampled during operations. The availability of a methodology to perform such signal validation in real-time or close to real-time would be a great asset in the nuclear industry. It would certainly help site engineers to identify actual and potential sensor failures or even system failures during normal operations.

For signal validation and process modelling, use of neural networks [1] has shown some significant advantages over traditional methods, like physical modelling methods, identification methods, time-series methods, and empirical methods. This is mainly because the neural network does not require much information about the process or signal such as process nonlinearities and process order, mathematical or physical relations among the variables of the given process. In addition, since the neural network is usually composed of nonlinear processing units (called neurons), it is capable of handling various nonlinearities in the process or signal. Some efforts to apply neural networks have been made in the nuclear industry. Several typical publications are [2,3,4]. In [2], the multi-layer feedforward neural network has been used to model processes from both a commercial power plant and the Experimental Breeder Reactor-II. Neural networks have been used in conjunction with physical process models to investigate anomalies in the main coolant system of the Japanese Advanced Thermal Reactor in [3]. In [4], use of neural networks for developing models of signal processing has been reported. However, the above mentioned work in applying neural networks to nuclear power plant relates to off-line analysis and assessment. Not much work has been done in real-time or close to real-time applications of neural networks, in dynamically validating signals in the nuclear industry. In the first part of this paper, neural network applications in a CANDU reactor, particularly in the Pressure and Inventory Control (PIC) system of such a reactor, will be presented.

Recently, extensive attempts have been made to apply the Artificial Intelligence (AI) technology, such as expert systems (ES), to various engineering problems. In particular, in the nuclear industry, a methodology has been proposed for the coupling of pre-trained artificial neural networks (ANN) to a rule-based expert system to perform monitoring and diagnostics of large scale systems [5]. Based on a decision tree, a knowledge-based method for predicting the robustness of a power system to severe disturbance has been presented in [6]. In those applications, the expert systems were based upon the commonly used relational database storing knowledge. None of them has utilized the principle of frames [7] to represent knowledge. In this paper, instead of using semantic networks or production rules, frames will be utilized for representing knowledge of a physical process in knowledge-based expert systems. A successful application of frames to an engineering problem, computer control in manufacturing systems, has been reported in [8].

Signal Validation Using Neural Networks

Usually, signal validation, automated by using computers and properly developed software, is composed of two parts: one, the core part of the system, which actually performs signal prediction and comparison, and the other which makes decisions about the validity of signals to be verified, which is basically an expert system. In this section, a detailed description of a signal validation approach will be given while another part, i.e. the expert system, will be discussed in the next section. Basically, the approach to be discussed in this section is a combination of the neural network with some signal processing principles [9,10,11].

The first part of this approach for signal validation is mainly composed of three components. The first one is development of a set of neural network models which can be used to provide reasonably accurate estimates for signals. More detailed description for the development will be given later in this section.

The second component of the signal validation approach is a Generalized Consistency Checking for Multivariables (GCCM) algorithm [9,10]. The implementation of the GCCM algorithm mainly consists of two steps. One is a systematic comparison of redundant sensor measurements with their predicted values at each time instance. The predicted values are generated by using a set of well trained neural network models. If any direct measurement and its predicted value are inconsistent, then the Cumulative Inconsistency Index (CII) of the direct measurement and all input signals, which are used to calculate the output prediction, are increased by one. This is repeated for all direct measurements of the output variable and CIIs are calculated for all the redundant measurements of each participating variable. A degraded signal will have the maximum possible CII regardless of the status of other measurements. The second step in performing the GCCM is to run through a complex logic to isolate the faulty signal. After completing simultaneous checking within one redundant measurement set, and cross-checking among the redundant measurement set of process variables, the complex logic is able to detect and isolate bias and calibration errors, and even the more complicated common-mode failure. Since this paper is focused on the neural network and expert systems, the reader is directed to [9,10] for further details on GCCM.

As the last component in the approach for signal validation, the whiteness test [11] and the sequential probability ratio test (SPRT) [9,10] of GCCM residuals, will be conducted. The residual here is the difference between actual measurements and GCCM estimates. The purpose of doing this is twofold: firstly, to detect possible signal degradations occurring during normal operations, and secondly, to verify how good GCCM estimates of signals are and to get hints on improving neural network models if it is necessary. Again, details of the whiteness test and SPRT are omitted due to the scope of the paper. References [9,10,11] are recommended for more details.

Neural Network Approach

The neural network may be viewed as a three-stage system as depicted in Fig. 1. The "receptors" convert stimuli from the external environment to the neural net. Central to the system is the "neural net", representing the brain, which usually consists of a massive network of neurons arranged in a highly elaborate structure with very complex interconnections. A neuron receives its inputs from either receptors or other neurons, or even effectors. The "effectors", on the other hand, convert the internal signals emitted by the neural net into responses as system outputs.

The neural network used in this paper follows the structure described in Fig. 2, which is a three-layer feedforward network. Corresponding to the three-stage system in Fig. 1, the three layers are called the input layer (i.e. the bottom layer), the hidden layer (i.e. the middle layer), and the output layer (i.e. the top layer), respectively. The data flow of the network is from the input layer to the output layer, i.e. from the bottom to the top in Fig. 2. Every neuron is a nonlinear mapping from multiinputs to the single-output expressed as:

$$y_j^k = f\left(\sum_i \omega_{ij}^k y_i^{k-1} + \theta_j^k\right) \qquad (1)$$

where $\{\omega_{ij}^k\}$ are weights connecting the output of the ith neuron at layer k-1 to the ith input of the jth neuron at layer k. $\{\theta_j^k\}$ is the threshold of the jth neuron at layer k. $\{y_i^{k-1}\}$ and $\{y_j^k\}$ are the output of the ith neuron at layer k-1 and the output of the jth neuron at layer k, respectively. The nonlinear function $f(.)$ has the following sigmoidal form

$$f(x) = \frac{1}{1 + e^{-\beta x}} \qquad \textit{with } 1 > \beta > 0. \quad (2)$$

Usually, β is selected from 0.2 to 0.9. The nonlinearity of the function increases as the β increases. The basic idea of neural networks is to estimate all weights ω_{ij}^k and thresholds θ_j^k in such a way that the ANN output approximately equals the desired value of the process output in the sense of certain optimization.

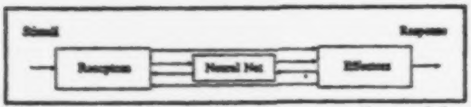


Figure 1 Diagramatic representation of the neural network

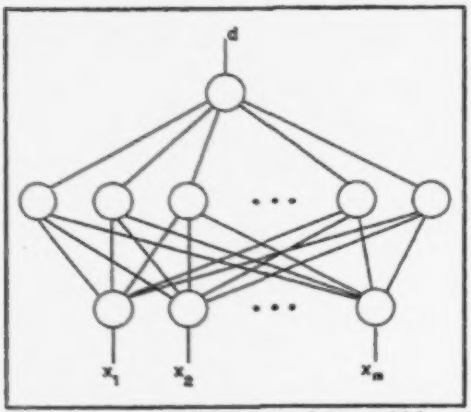


Figure 2 A configuration of the multi-layer feedforward neural network

Learning Algorithms

The manner in which the neuron of a network are organized is intimately linked with the learning algorithm used to train the network. For the multi-layer feedforward neural network to be discussed in this paper, the learning algorithm is selected as the "back-propagation algorithm" [1].

Suppose the error signal at the output of cell k at iteration n is defined by

$$e_k(n) = d_k(n) - y_k(n) \qquad (3)$$

where $d_k(n)$ is the desired response for computing $e_k(n)$ at iteration n and $y_k(n)$ refers to the calculated signal appearing at the output of cell k at iteration n. Hence, the mean square error at layer j is (except for a scaling factor)

$$\begin{aligned} \mathcal{E}(n) &= \sum_j e_j^2(n) \\ &= \sum_j (d_j(n) - y_j(n))^2. \end{aligned} \qquad (4)$$

Minimization of Eq.(4) is performed in two phases of computation, the forward phase and the backward phase. In the forward phase, the connection weights and thresholds remain unchanged throughout the network, and the output signal of every cell in the network is computed on a cell-by-cell and upward layer-by-layer basis. On the other hand, the backward phase of computation starts at the output layer by passing the error signals downward through the network, layer-by-layer, and recursively computing the δ for each cell. The connection weights and thresholds of the network are modified during this phase in accordance with the modified delta rule derived by minimizing Eq.(4). Detailed derivation of the algorithm can be found in [1]. For the sake of completeness, however, the outlined back-propagation algorithm is given as follows:

Step 1. Normalize the given input-output patterns into the range from 0.1 to 0.9;

Step 2. Assign all weights and thresholds in the ANN to a random real number, within the range of [-1.0, +1.0];

Step 3. Present the network with input vectors $x(n)$ and desired response $d(n)$ for iteration n=1, 2, ... The vector $x(n)$ is defined as:

$$x(n) = [x_1(n), \ldots, x_m(n)] \qquad (5)$$

where m stands for the number of input variables to the process;

Step 4. Compute the output of the network by proceeding forward through the network, layer by layer:

$$y_j^k(n) = \frac{1}{1 + e^{-\beta net_j(n)}} \qquad (6)$$

with $net_j(n)$ is given by

$$net_j(n) = \sum_i \omega_{ij} y_i^{k-1}(n) + \theta_j^k. \tag{7}$$

If cell j is in the bottom (input) layer, then

$$y_i^{k-1}(n) = x_i(n) \tag{8}$$

if cell j is in the top (output) layer, then

$$y_j^k(n) = O_j(n) \tag{9}$$

where $O_j(n)$ is the jth element of the output vector. Hence, compute the error signals by

$$e_j(n) = d_j(n) - O_j(n) \tag{10}$$

for all j. In this paper, $j=1$ since the ANN model has always one output variable;

Step 5. Calculate the δ's of the ANN by proceeding backward, layer by layer. For a cell j in the output layer, compute

$$\begin{aligned}\delta_j^k(n) &= \beta O_j^k(n)\,(1 - O_j^k(n))\\ &\quad \times (d(n) - O_j^k(n))\end{aligned} \tag{11}$$

for $j=1$. For an arbitrary hidden unit j, compute $\delta_j^k(n)$ by using

$$\begin{aligned}\delta_j^k(n) &= \beta y_j^k(n)\,(1 - y_j^k(n))\\ &\quad \times \sum_l \delta_l^{k+1}(n)\,\omega_{jl}^{k+1}(n)\end{aligned} \tag{12}$$

for all j. Hence, adjust the weights according to the modified delta rule:

$$\begin{aligned}\omega_{ij}^k(n+1) &= \omega_{ij}^k(n)\\ &\quad + \alpha\,(\omega_{ij}^k(n) - \omega_{ij}^k(n-1))\\ &\quad + \eta\,\delta_j^k(n) y_i^k(n)\end{aligned} \tag{13}$$

for all i and j. In Eq.(13), α and η are the momentum term and the learning rate, respectively. Usually, these two should be selected within the range of 0 and 1. While α determines the effect of past changes on the current direction of movement in weight space, η specifies the length of adjustment made in the weights at each pass. The internal cell thresholds are adapted in a manner similar to the weights by treating them as connection weights fed with constant-valued inputs. That is

$$\begin{aligned}\theta_j^k(n+1) &= \theta_j^k(n)\\ &\quad + \alpha\,(\theta_j^k(n) - \theta_j^k(n-1))\\ &\quad + \eta\,\delta_j^k(n).\end{aligned} \tag{14}$$

Step 6. Repeat the computation by going back to Step 3 until the following inequality is satisfied:

$$\sqrt{\frac{1}{N}\sum_{p=1}^{N}(d^p(n) - O^p(n))^2} < \varepsilon \tag{15}$$

where N is the number of patterns to be trained, and $d^p(n)$ and $O^p(n)$ are the pth desired response and the predicted response at iteration n, respectively. In addition, ε is a small number to control the training accuracy of the network.

In this paper, the number of nodes in the input layer corresponds to the number of input variables associated with the output variable. The output layer of the neural network contains only one neuron which indicates that there is only one output variable in the process. For all ANN models developed in this paper, only one hidden layer including seven nodes has been selected. According to our experience and previous work reported in the other papers, different selections in the number of hidden layers and their corresponding number of nodes do not improve training results significantly.

Knowledge-Based Expert Systems

In this section, a knowledge-based expert system will be presented. As one of the two key factors in developing an expert system, the scheme utilized for representing knowledge will be discussed first. And then, a Prolog-based reasoning process will be proposed.

Knowledge Representation

Knowledge is information based on which decisions can be made through an expert system. The knowledge should be represented in such a way that the knowledge base has some desired features. There are different methodologies available to represent knowledge (called knowledge representation i.e. KR.) The type of KR which is appropriate for a given situation mainly depends on two factors. One is the nature of knowledge, and another is the way knowledge is to be applied.

To develop an expert system for real-time or close to real-time signal validation in a complex physical process, like the Pressure and Inventory Control (PIC) system in a CANDU reactor at the Point Lepreau Generating Station (PLGS), two issues are particularly important. They are the size of the knowledge base and the speed of expert systems to make decisions, respectively. The reality of the expert system to be discussed in this paper is that the potential size of the knowledge base will become bigger and bigger. In addition, the nature of the safety operation requirement in a nuclear power plant requires that the expert system is allowed to spend only a small amount of time to make decisions about the validity of signals. Some commonly used KR schemes, such as semantic networks and production rules, become unmanageable as the size of the knowledge base increases [12]. Also, the interface between the knowledge base and inference engine becomes more difficult. Thus, the resultant expert system may not be appropriate, or less efficient, in a time-critical system, like the system discussed in this paper.

The frame-based KR [7,8,12] is the area of artificial intelligence that has used object-oriented systems to the greatest extent to date. A frame is a KR scheme that associates an object with a collection of features, such as facts, rules, defaults, and active values. Each feature is stored in a slot. Therefore, the body of a frame is composed of a set of slots related to a specific object. A slot may contain intrinsic features such as the object name (i.e. the frame name), attribute and values, attribute with default values, rules or procedures to determine values, pointers to related frames, and information about the frame's parent.

A typical frame introduced in this paper is shown in Fig.3. The name of the frame is "FFORWARD-NN", and that of its parent frame name is "ANN." There are four slots in the frame. They are named "prediction", "ann_input", "ann_output", "ann_model", respectively. As indicated by the names, these slots hold values of ANN predictions, input and output information for the ANN training, and properly trained ANN models. Two predicates are attached to the slots. These predicates behave as daemons, constantly monitoring the situation, and whenever a predefined situation arises, they will automatically carry out the appropriate function. One predicate attached to slot "prediction" is to compute ANN predictions when the new value of predictions is activated. Another is attached to the rest of three slots. When the values of all three slots are activated, the predicate will be implemented, i.e. the ANN training procedure will be carried out.

Since frames are connected to each other by a frame tree that describes their hierarchical structure, a reasonable hierarchical structure can ease the use of the frames especially during the reasoning process. The connection of the frames designed in this paper is shown in Fig.4. From the figure, it can be seen that the connection is a hierarchical structure and there are four levels in the structure. The top level is the root frame, called "CASE-FRAME" which holds very general slot values, like case information, initial constants, and sampling frequency etc. The frame "PROCESS-FRAME", whose slots hold values about all process, variable, and signal related information, is related to its parent frame "CASE-FRAME" by the "similar" relationship since the entity sets they represent have enough properties in common. The frames "GCCM", "SPRT", "RESIDUAL-CHECK", and "ANN" are in level 3 and related by the "sibling" relationship to the frame "PROCESS-FRAME." This is because the information stored in "PROCESS-FRAME" are needed when the four frames perform their calculations. The relationship by which these four frames are related to the frames at the bottom of the hierarchy is called "subset", or "isa." The frame "FFORWARD-NN," for example, is a ("isa") particular case of frame "ANN." A similar relationship exists for frames "M-SPRT", "V-SPRT", "MV-SPRT", and "CORRELATION."

Reasoning Process

The reasoning process is one of two important parts to construct an expert system. In this paper, the reasoning process is performed by using a set of rules which use the knowledge stored in the frames. The entire logic of the reasoning process is very complex. Due to the length limitation of the paper, only one part of the reasoning process will be given in this sub-section. To obtain ANN predictions, a sample rule written in Prolog is given as follows:

```
ann_prediction(Ann_type) :-
    Ann_type = fforward, !,
    get_value(fforward, input_sig_ndx,Sig_ndx),
    get_value(fforward, ann_model, Ann_model),
    get_prediction(fforword,Sig_ndx,Ann_model,
                   Prediction),
    write_new_value(Prediction).
ann_prediction(Ann_type) :-
    Ann_type = fforward,
    output_error_message(Ann_type), !.
```

The first predicate is used to compute ANN predictions for given a set of input signals and an ANN model. The second predicate is designed to produce error messages if the first predicate fails to obtain the required ANN prediction.

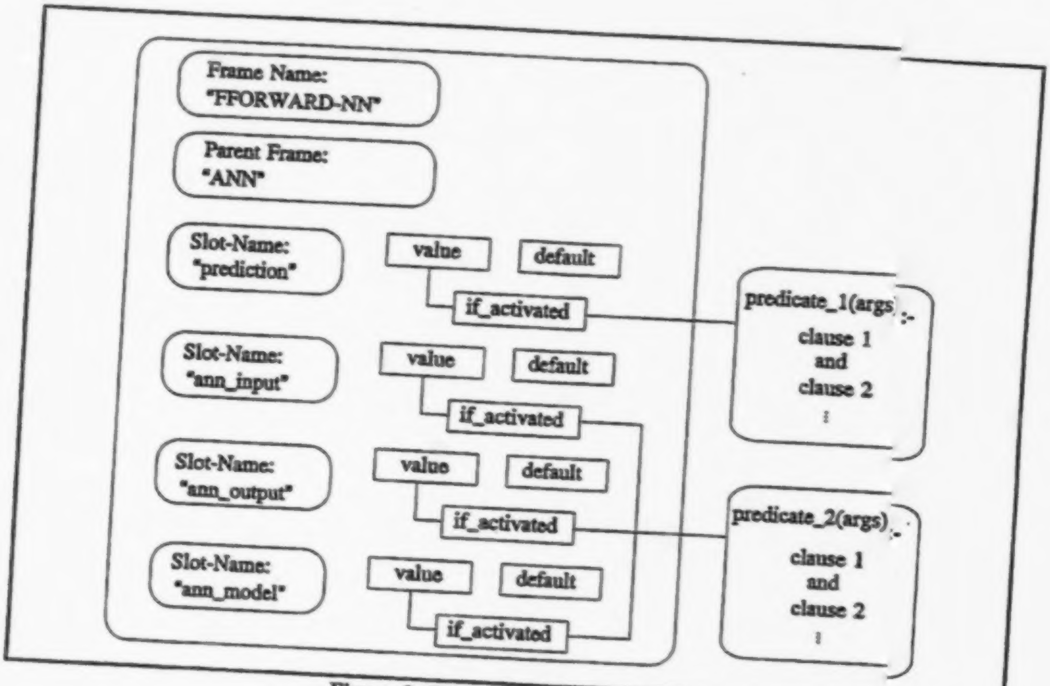

Frame Name:
"FFORWARD-NN"
Parent Frame:
"ANN"
Slot-Name:
"prediction"
value
default
if_activated
Slot-Name:
"ann_input"
value
default
if_activated
Slot-Name:
"ann_output"
value
default
if_activated
Slot-Name:
"ann_model"
value
default
if_activated
predicate_1(args) :-
clause 1
and
clause 2
predicate_2(args) :-
clause 1
and
clause 2

Figure 3 Frame FFORWARD-NN

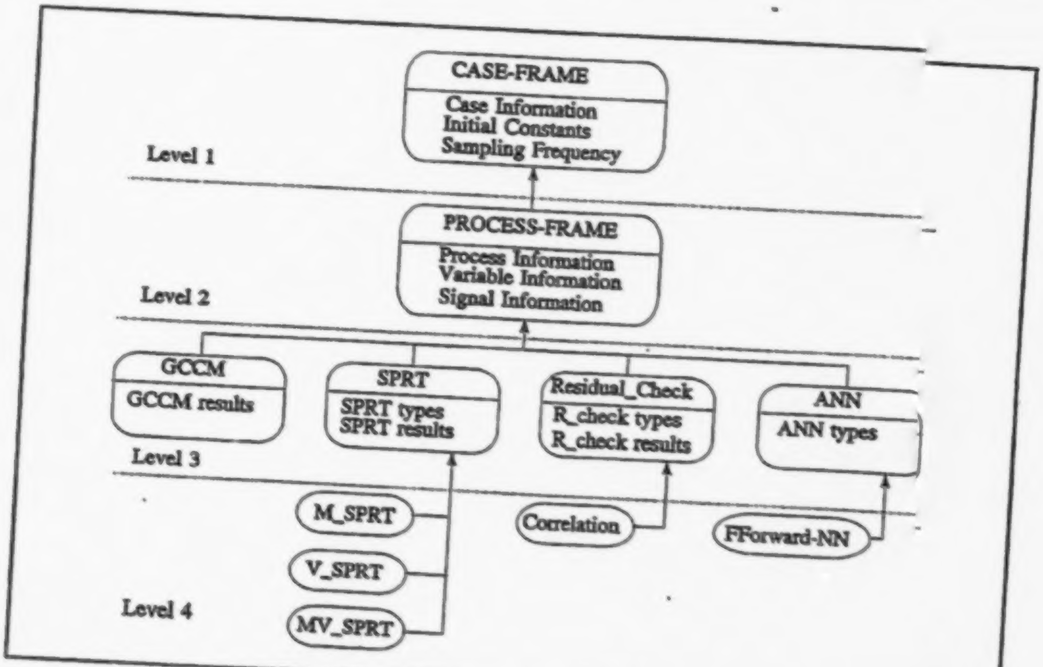

CASE-FRAME
Case Information
Initial Constants
Sampling Frequency
Level 1
PROCESS-FRAME
Process Information
Variable Information
Signal Information
Level 2
GCCM
GCCM results
SPRT
SPRT types
SPRT results
Residual_Check
R_check types
R_check results
ANN
ANN types
Level 3
M_SPRT
V_SPRT
MV_SPRT
Correlation
FForward-NN
Level 4

Figure 4 Hierarchical representation of frames for knowledge representation

Results

In this section, only some of the results related to the system will be provided, since some portions of the system are still under development.

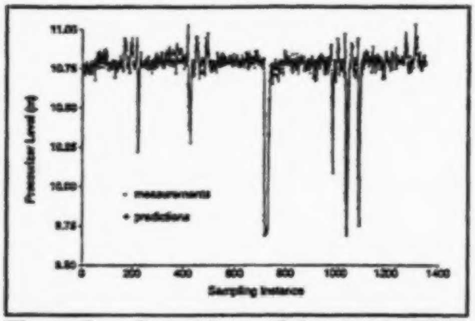


Figure 5 Comparison of the signal (GWX AX 150), sampled in Dec. 1992, and the prediction of the neural network model trained for GWX AX 150 in Dec. 1992.

In the PIC system, eight variables, which are considered to be closely related to each other, have been selected among other variables at PLGS. The data of the eighteen signals associated with the eight variables were collected from December 7 to December 11, 1992. The sampling interval is five minutes. The total number of sampling points is 1350. Three ANN models have been developed to describe three sub-systems associated with the PIC system: pressurizer level, pressurizer liquid temperature, and pressurizer pressure, respectively. Details of the ANN model for the pressurizer level are given in Table I where the plant identifiers of signals associated with corresponding variables and used for the ANN training are given in the brackets. The learning rate η, momentum term α, and the constant β in the sigmoidal function are all selected as 0.9. The training error in the table is based on Eq.(15), that is, the standard deviation. A detailed discussion on the modelling process can be found in another paper by the same authors [13].

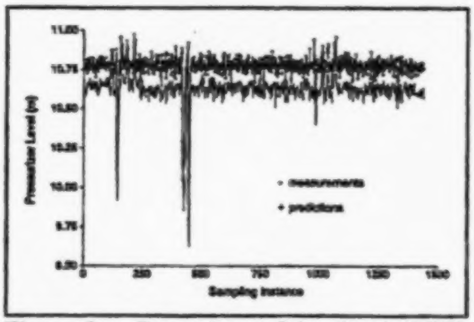


Figure 6 Comparison of the signal (GWX AX 150), sampled in Mar. 1993, and the prediction of the neural network model trained for GWX AX 150 in Dec. 1992.

Comparison of the neural network predictions with the associated measurements is provided in Figure 5. Notice that the first 800 predictions are produced by using the training data, whereas predictions after the 800th sampling instance are calculated by using the data which was not used for training. Figure 6 shows comparison of the ANN prediction with an equivalent set of real-time data generated at PLGS from March 22 to March 26 in 1993. It can be seen that the developed ANN model can accurately predict the variables even during transient operations like fuelling, and various types of system testing. However, the predicted outputs are somewhat shifted. This may be due to changes in the working conditions of the reactor. Another possible explanation is that the neural network model needs further improvement so that the working

Table I: ANN model for Pressurizer Level (GWX AX 150) in PIC system

Input variables: (with plant identifier)	Reactor Inlet Header Temperature (GWX AI 1224); Heat Transport Bleed Flow (GWX AI 1130); Heat Transport Feed Flow (GWX AI 1132); Pressurizer Liquid Temperature (GWX AI 2417); Pressurizer Pressure (GWX AI 1160); Pressurizer Variable Heater Input (GWX AI 2450); Calculated Linear Power (GWX AX 4);
Training patterns:	from No.1 to No.800
Number of iterations:	12739
Training error:	0.030365

condition change can be detected.

Signal validation has been performed by following the approach discussed in the previous sections. The results have confirmed the measurements sampled during normal operations in the CANDU at PLGS. It indicates the validity of the proposed system using both ANN models and the frame-based knowledge base for signal validation.

Conclusions

Signal validation of the PIC system in the CANDU at PLGS has been performed by using a knowledge-based expert system. Several ANN models have been developed and used to accurately predict variables of interest, during normal operations, including either static operations or transient operations, such as fuelling, and various types of system testing. The frame methodology has been utilized for the knowledge representation to construct the knowledge-based expert system for signal validation. As a consequence of representing knowledge in frames, the size of the knowledge base is considerably reduced as opposed to the semantic network and production rule approaches. Therefore the expert system may be exercised at a frequency consistent with the potentially rapidly changing processes in a CANDU reactor. The knowledge representation and reasoning process are implemented in Prolog, whereas the numerical algorithms are written in C. The approach presented in this paper shows strong, solid potential for real-time applications of signal validation in the nuclear industry. It should be straightforward to apply the method to problems in the other industries.

References

1. D.E. Rumelhart and J.L. McClelland, *Parallel Distributed Processing*, *The MIT Press*, vol. 2, Cambridge, Massachusetts, 1986.
2. E. Eryurek and B.R. Upadhyaya, "Sensor validation for power plants using adaptive backpropagation neural network," *IEEE Trans. on Nuclear Science*, vol.37,pp.1040-1047,1990.
3. K. Watanabe and K. Tamayama, "Diagnosis method utilizing neural networks," appear in *Artificial Intelligence in Nuclear Power Plants*, P.J. Haapanen (ed), vol. 2, pp.137-148,1990.
4. M.-S. Roh, S.-W. Cheon, and S.-H. Chang, "Power prediction in nuclear power plants using a back-propagation learning neural network," *Nuclear Technology*, vol.94,pp.270-278,1991.
5. L. Tsoukalas and J. Reyes-Jimenez, "Hybrid expert system-neural network methodology for nuclear plant monitoring and diagnostics," *Applications of Artificial Intelligence VIII*, SPIE vol. 1293, pp.1024-1030,1990.
6. B. Jeyasurya and S.S. Venkata, "A knowledge-based approach for power system dynamic security assessment," *The 3rd Int. Conf. on IEA/AIE*, South Carolina, pp.645-652,1990.
7. M. Minsky, "A framework for representing knowledge," in *Reading in Knowledge Representation*, R.J. Brachman and H.J. Levesque eds., Morgan Kaufmann Publishers, Inc., pp.245-262,1985.
8. R. Lingarkar, L. Liu, M.A. Elbestawi, and N.K. Sinha, "Knowledge-based adaptive computer control in manufacturing systems: a case study," *IEEE Trans. on Systems, Man, and Cybernetics*, vol.20,no.3, pp.606-618,May/June, 1990.
9. *Development of an Integrated Remote Monitoring and Diagnostic System (IRMAD)*, Final report to Canadian Space Agency, S.S.C. File no. 09SC.9F006-1-ACSI, vol.1, 1993.
10. O. Glöckler, B.R. Upadhyaya, V.M. Morgenstern, and J. Olvera, "Generalized consistency checking of multivariable redundant measurements and common-mode failure detection," *Department of Nuclear Engineering, University of Tennessee*.
11. N.K. Sinha and B. Kuszta, *Modeling and Identification of Dynamic Systems*, Van Nostrand Reinhold Company, New York, 1983.
12. D. Partridge, "The scope and limitation of first generation expert system," in *Future Generation Computer System*, Amsterdam: North Holland, pp.1-10, 1987.
13. H. Dai and J.W. Thompson, "Use of neural networks for process modelling and signal validation of nuclear power plant," *5th UNB Artificial Intelligence Symposium*, Fredericton, N.B., pp.3-13, August 12-14, 1993.

Acknowledgement

The work reported on herein was initially sponsored by the Government of New Brunswick/ACOA and the Canadian Space Agency under STEAR program.

374968

Control of Multi-DOF Robots Using Neural Networks

Kazuo Kiguchi[1] **& Dan Necsulescu**[2]
Dept. of Mechanical Engineering
University of Ottawa
Ottawa,Ontario,K1N 6N5

#149758

Abstract

Currently, numerous Neural Networks studies have been performed in order to apply Neural Networks' ability of acquiring and storing experiential knowledge and adjusting robot controllers to new environments. Neural Networks can be applied to robot controllers directly or indirectly. It is known that the nonlinear Neural Networks controllers are able to take into account the nonlinear effects of the plant. This ability of Neural Networks is very useful for controlling robot manipulators which are nonlinear systems. This might be particularly useful for position/force control of multi-DOF robots. Generally, the Neural Networks of the PDP model can only realize static mapping by itself and can not incorporate the dynamics necessary for an advanced controller. In order to realize dynamic mapping, past input/output information of the plant is used as input signals for the backpropagation type of Neural Networks.

In this paper, the nonlinear three-layer Neural Networks using a dynamic back-propagation method which based on the output signals of the hybrid controller and past input/output information of the plant is proposed for the Cartesian position/force control of multi-DOF robots. The generalized delta law is used as a learning law to adjust the weights of the Neural Networks at every sampling time.

Experiments were done with a 3-DOF Direct-Drive Planar Robot Manipulator. The Neural Networks Controller controls the position and the angle of the end-effector as well as the force applied to an unknown environment. The experimental results show the effectiveness and flexibility of the Neural Networks Position/Force Controller for Multi-DOF Robots.

Introduction

A neural network, an artificial neural network in this paper, is motivated by the research of the human brain and its nervous system. Neural networks consist of many synapses and neurons. The human brain contains roughly 10^{11} neurons and 10^{15} synapses. Neural networks are able to store the pattern or function information in themselves. There are many kinds of neural network architectures and learning laws. Although neural networks were studied for a long time since McCulloch and Pitts proposed the first neuron model in 1943 [1], they are still subject of research because the understanding of biological neural networks is not developed enough. Currently applications to Robotics have been studied in order to apply neural networks' ability of acquiring and storing experiential knowledge and adapting to new environment. In Robotics, neural networks can be used for realization of forward or inverse kinematics and dynamics, as well as for control and identification of the system.

Application of neural networks to control is first stated by Widrow [2] in 1960s. Afterwords Kawato et al. first applied neural networks to robot control [3][4] in 1980s, many other researchers have applied neural networks to robot control in various ways. Narendra and Parthasarathy proposed the identification and control of dynamical systems using neural networks and showed simulation results [5]. They discussed static and dynamic back-propagation methods for the adjustment of parameters. Yabuta and Yamada proposed neural networks for force control of 1DOF robot [6][7] and tested it experimentally. They studied also the stability of the linear neural network controller using a Lyapunov function [8]. Fukuda et al. applied neural networks for hybrid position/force control [9-11]. In this controller, neural network is used to self-tune the PID parameters. Their simulation results and experimental results show that the controller is robust to environment dynamics and manipulator configuration changes. Furthermore, they used neural networks to control a manipulator, approach the environment efficiently and to recognize the contact with the environment.

---

[1] phone (613) 564-9292.
[2] phone (613) 564-5761.

The hybrid position/force control which was proposed by Raibert and Craig in 1981 [12] is one of the most important contributions in the position/force control. The basic idea is separate direction for force control, that is normal to the constraint surface, and for the position control, which is orthogonal to the constraint surface in Cartesian coordinate system. The Cartesian forces are eventually transformed into joint torques. Since it is not very difficult to design the hybrid controller, we decided to build this controller first and use the outputs from the hybrid controller for the input signals of the neural network controller.

In this paper, the direct neural network Cartesian controller (using the outputs of the hybrid controller) is proposed for a 3DOF planar robot and is experimentally tested.

Hybrid Control

In order to apply the desired force to the environment with the robot manipulator shown in Fig.1, y face component and the angle of the end-effecter with regard to the environment have to be controlled. For this purpose, a hybrid position/force controller seems to be a proper controller.

The dynamics equation of the planar robot is written as

$$M(q)\ddot{q} + h(q,\dot{q}) + Cc \cdot sgn(\dot{q}) = t - J^T fd \qquad (1)$$

Because of low velocity, we assume viscous friction is negligible. Here M is the inertia matrix, h component is Coriolis and Centrifugal components, Cc is Coulomb friction, t is output torque, J is Jacobian fd is the desired force applied to the environment and **q** is a angular position vector.

From the equation (1), the required motor torque equation results as

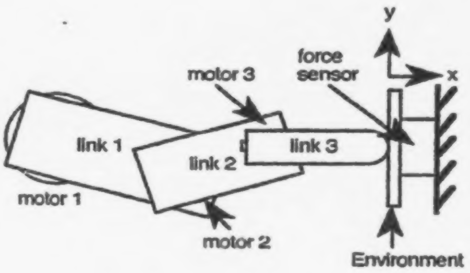


FIGURE.1: Robot Manipulator

$$t = M(q)\ddot{q} + h(q,\dot{q}) + Cc\, sgn(\dot{q}) + J^T fd. \qquad (2)$$

The acceleration of the end-effector in Cartesian coordinate is

$$\ddot{x} = J\ddot{q} + \dot{J}\dot{q}. \qquad (3)$$

Here x is a Cartesian position vector.

From the equation (2) and (3) and the selection matrix S, the following equation can be written for hybrid control [12]

$$\begin{aligned} t = M(q)J^{-1} [(I - S)ux - \dot{J}\dot{q}] + h(q,\dot{q}) \\ + Cc\, sgn(\dot{q}) + J^T fd + J^T S\, uf. \end{aligned} \qquad (4)$$

Here ux is a command vector for position control and uf is a command vector for force control. For resolved acceleration method, ux and uf can be obtained from linear control laws:

$$ux = \ddot{x}d + K_{pv} (\dot{x}d - \dot{x}) + K_{pp} (xd - x) \qquad (5)$$
$$uf = K_{fp}(fd - f) + K_{fi} \int (fd - f)\, dt. \qquad (6)$$

Where

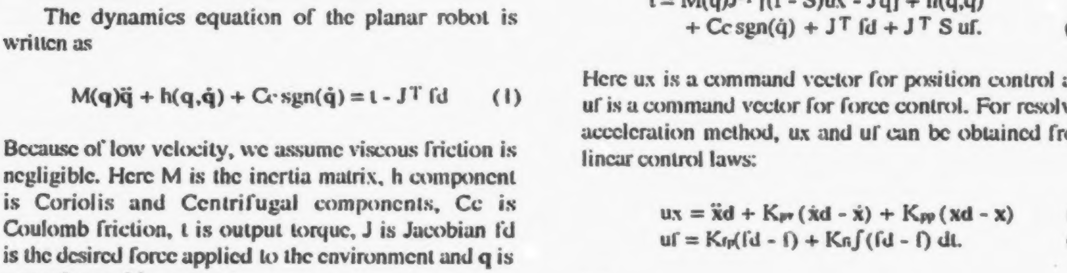


Figure 2: Block Diagram of Hybrid Controller

K_{pp} : Position Gain for Position Control
K_{pv} : Velocity Gain for Position Control
K_{fp} : Position Gain for Force Control
K_{fi} : Integration Gain for Force Control.

The block diagram of the hybrid controller is shown in Fig.2. Using this controller, the force applied to the environment in y-direction and the angle of the end-effector can be controlled.

Obtaining the desired position or force with the hybrid controller is very difficult for several reasons. First, the model is not complete and exact. Second, we can not use high gain for force control for contact with an unknown environment. Third, it is difficult to compensate friction; as far as the robot manipulator is concerned, the friction force depends on configuration of the robot manipulator. Fourth, it is impossible to avoid external disturbance force completely. Furthermore, the properties of some environment such as rubber, plastic, styrofoam, etc. are affected by temperature and humidity, so these properties might change with time. In order to solve these problems, some adaptive scheme has to be implemented.

Neural Network Controller

In order to obtain the desired position and force, the neural network controller seems to be a suitable controller because of its ability of adapting to an unknown environment and compensating the nonlinearities of the robot. There are several kinds of applications to robot controller, as presented in the introduction. The direct Neural Network (N.N.) robot controller (Fig.3) seems the most effective way of application. However, since the conventional controller, like hybrid controller, can be designed without much difficulty, such a controller will be analysed first. Afterwards the outputs of this controller will be used as inputs of neural network controller (Fig.4). Furthermore, if we use the output of the hybrid controller as a part of input to the robot (Fig.5), we can expect that the neural network compensates the disturbance and model error faster.

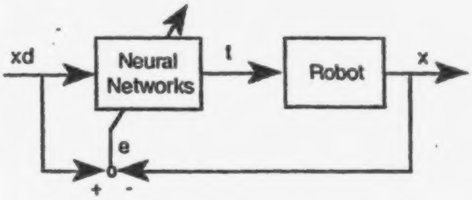


Figure 3: Direct N.N. Controller

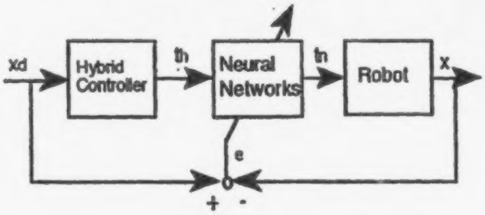


Figure 4: Modified direct N.N. Controller

For this controller, the neural network compensates friction, model error, other external disturbances and therefore the hybrid controller used for this control does not need to compensate friction. The equation (4) can in this case be changed as

$$\tau = M(q)J^{-1}\,[(I - S)ux - \dot{J}\dot{q}] + h(q,\dot{q}) + J^{T} fd + J^{T} S\,uf. \qquad (7)$$

The command vectors for this controller results as

$$ux = \ddot{x}d + K_{pv}(\dot{x}d - \dot{x}) + K_{pp}(xd - x) \qquad (8)$$

$$uf = K_{fp}(fd - f). \qquad (9)$$

As we can see in equation (9), the integral of the error part is not needed any more because the neural network

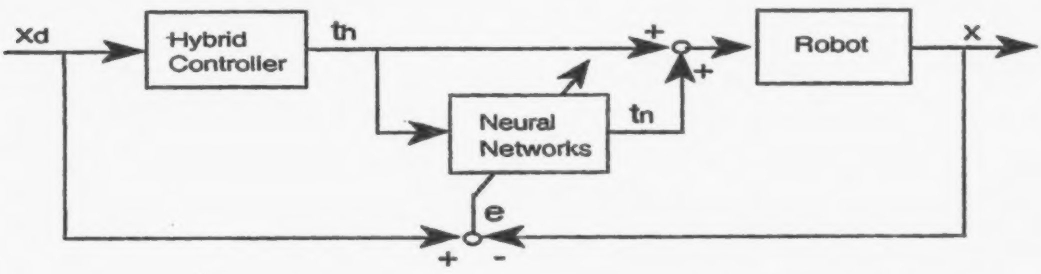


Figure 5: Direct N.N. Controller used in this paper

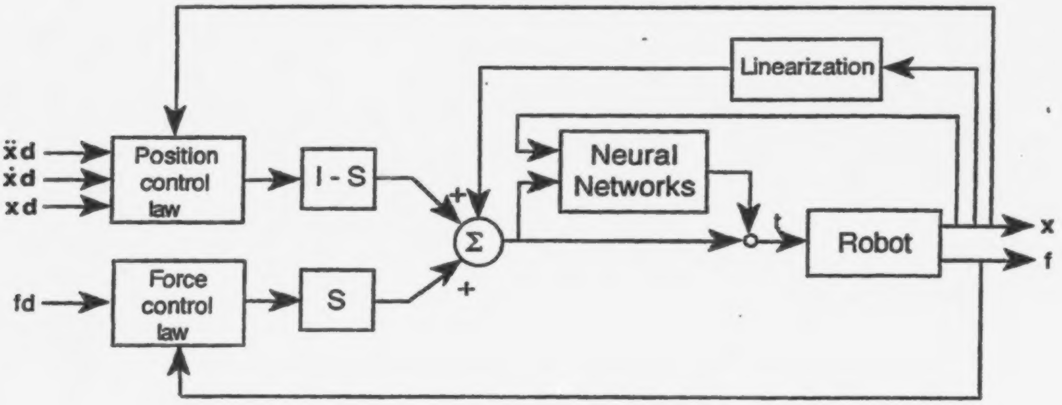


Figure 6: Neural Network Controller

controller compensates the error of the hybrid controller. Furthermore, the gain in the equation (9) does not need to be changed for various environments since the neural network controller adjusts itself to unknown environments.

The detail of the neural network controller is shown in Fig.6. Actually, this is a controller combining the hybrid controller shown in Fig.2 and the direct neural network controller shown in Fig.3.

Architecture of Neural Network

The architecture of the neural network is shown in Fig.7. Input signals to the neural network are current and previous position of the robot manipulator in Cartesian coordinates and the computed torque commands from the hybrid controller for each motors. The configuration (position) and the velocity of the robot manipulator are known for the current and previous position. The desired acceleration of the robot manipulator needed to obtain the desired position/force can be known from the computed torque commands from the hybrid controller. Output signals from the neural network are force commands for robot manipulator in Cartesian coordinates. These force commands are converted to torque commands for each motor using an equation which describes the relationship between joint torques and Cartesian forces

$$\tau = J^{T} f. \qquad (10)$$

The robot manipulator receives the sum of the torque commands from the controller. The result of the robot manipulator movement in response to the torque commands is measured by the encoders of each motors and by the force sensor. The weights of the neural network controller are adjusted every sampling time (1ms) using both position and force error. The back-propagation method is used as a learning law for the weights adjustment.

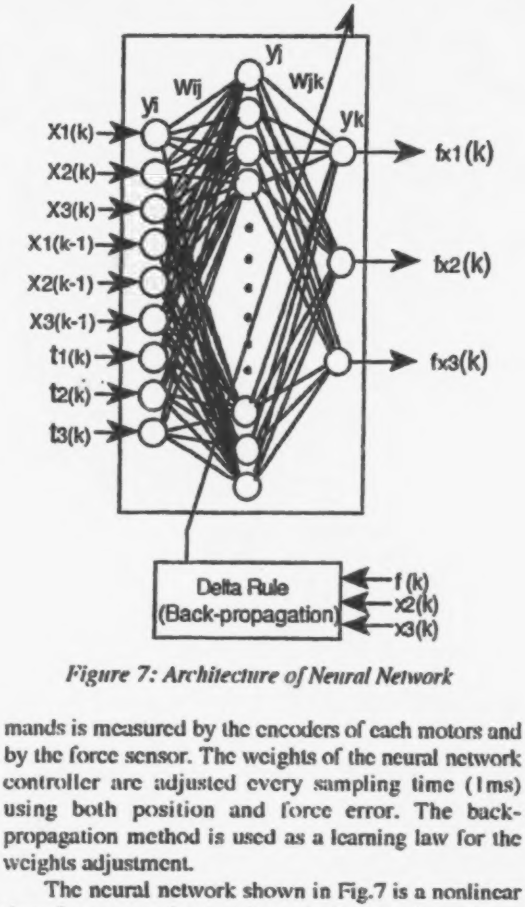


Figure 7: Architecture of Neural Network

The neural network shown in Fig.7 is a nonlinear three-layer neural network controller. The input layer consists of 9 neurons, the hidden layer consists of 12 neurons and the output layer consists of 3 neurons. Signals to the input layer are multiplied by weights wij and summed in each neuron in the hidden layer. The neuron is activated if summed value yields each threshold value. Output signals from the hidden layer are multiplied by weights wjk and summed in each neuron in the output layer. If the summed value yields each threshold value, the neuron is activated. Signals from the output layer are force commands for robot manipulators in Cartesian coordinates. The weights and threshold values of the neural network are adjusted every sampling time (1ms).

The activation function (sigmoid function) used in this neural network is written as

$$f(u) = 2/(1+e^{-u}) - 1 \tag{11}$$

where u is the input to the layer.

The back-propagation method is described as follows [13].

$$E = 1/2\Sigma(yd - y)^2 \tag{12}$$
$$uj = \Sigma(wijyi) \tag{13}$$
$$yj = f(uj) \tag{14}$$

where

E : Squared Error
yd : Desired Output from the Neural Network
y : Output from the Neural Network
uj : Input to the Neuron j
yj : Output from the Neuron j
wij : Weight for the signal from the Neuron i to the Neuron j.

The new weights of the neural networks can be written as

$$wij(k) = wij(k-1) + \eta S(ek(k)f'(uk)wjk(k-1))f'(uj)yi(k) \tag{15}$$
$$wjk(k) = wjk(k-1) + \eta ek(k)f'(uk)yj(k) \tag{16}$$

where η is a learning rate.

In this neural network controller, the error in the equation (12) is the difference between desired robot manipulator position/force and measured robot manipulator position/force. Therefore precise tracking of the desired position/force can be expected with this algorithm.

Experimental Set-Up

The experimental set-up consists of a 3-DOF Direct-Drive robot manipulator, motor drivers, a DSP and a force sensor. Fig.1 shows configuration of 3-DOF Direct-Drive robot manipulator. Fig.8 shows the experimental set-up. There are different type of motors in each manipulator joint. The motor1 is NSK 0810, the motor2 is NSK 0408 and the motor3 is Parker Hannifin System7 7521.250. The dSPACE is based on a DSP and include ADC, DAC and incremental encoder cards. The torque commands are sent to NSK motor drivers and System7 driver from dSPACE in which the control program is stored. The drivers realize motor commands. The force sensor (FS6-120A) sends a feedback signal to dSPACE. The notations in Fig.8 correspond to:

DSP - dSPACE
Motor 1 - NSK Megatorque Motor System 0810 (Max. 88.2 Nm)
Motor 2 - NSK Megatorque Motor System 0408 (Max. 9.8 Nm)
Motor 3 - Parker Hannifin Co. System7 Type 7521.250 (Max. 2 Nm)

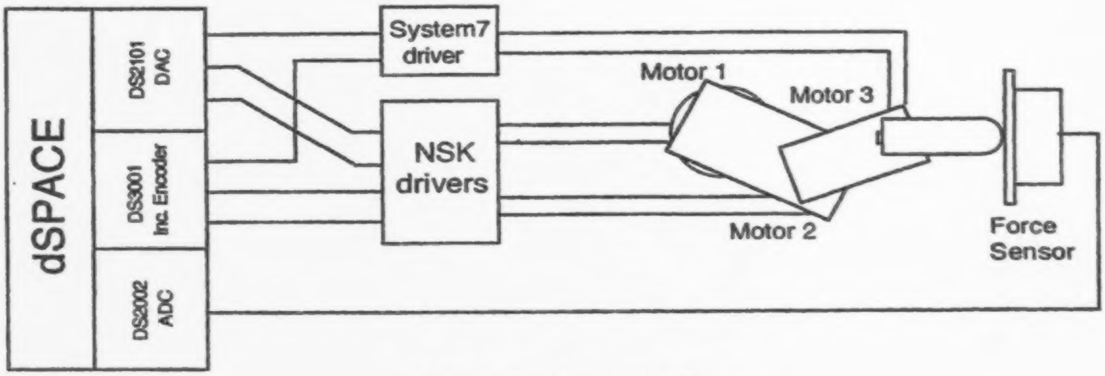


Figure 8: Experimental Set-Up

Force Sensor - Barry Wright Co.
6-axis force sensor FS6-120A.

Experimental Results

The neural network controller introduced in this paper controls the force applied to the environment, y position and the angle of the end-effector of the robot in Cartesian coordinate (see Fig.1). The purpose of this experiment is to verify the possibility and effectiveness of the neural network controller for pushing (force control) against an unknown environment. For the unknown environment, Styrofoam and Rubber are used. As explained before, the gains of the hybrid controller must be changed depending on the properties of the environment. However, the gains of this neural network controller need no explicit change since this controller adjusts them implicitly to the unknown environment. The gains in equations (8) and (9) of the hybrid control part of the neural network controller are

$$K_{pp} = 400.0\ ,\ K_{pv} = 80.0\ ,\ K_{fp} = 0.25.$$

In this experiment, these gains are used for both Styrofoam and Rubber environment.

Fig.9 shows the force response result of the robot manipulator for Styrofoam with the neural network controller about 10 seconds after the experiment is executed. The desired force (target force) is changed every 2.5 seconds between 10N and 20N. The learning rate η of the neural network is 0.00005. The results show that the robot manipulator follows well the desired force.

Fig.10 shows another experimental result for Styrofoam case. The desired force is a combination of a step signal, which is changed every 2.5 seconds between 10N and 20N, and a sinusoidal signal, $4\sin(\pi t)$. The learning rate is same as before. The result also shows the robot manipulator follows well the desired force.

Fig.11 shows the force response result about 10 seconds after the experiment is executed of the robot manipulator for Rubber case with the neural network controller. The desired force (target force) is changed every 2.5 seconds between 10N and 20N as before. The

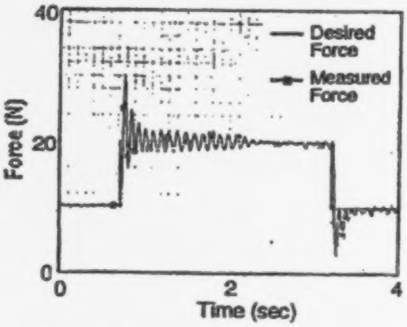


Figure 9: Measured Force Response 1

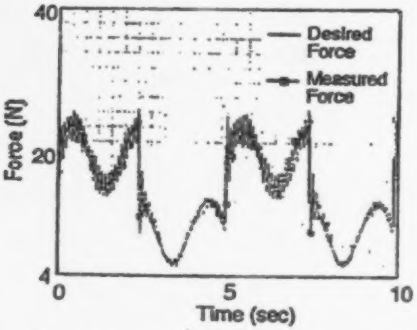


Figure 10: Measured Force Response 2

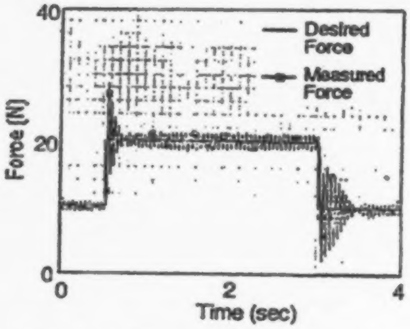


Figure 11: Measured Force Response 3

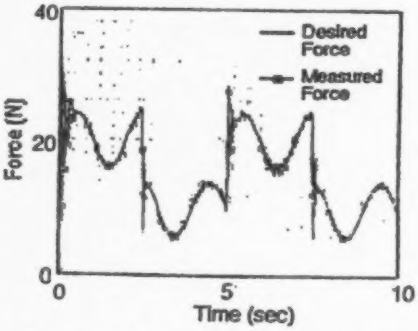


Figure 12: Measured Force Response 4

learning rate η of the neural network is 0.00005. This result also shows the robot manipulator follows well the desired force.

Fig.12 shows the beginning of another experimental result for Rubber case. The desired force is combination of step signal, which is changed every 2.5 seconds between 10N and 20N, and sinusoidal signal, $4\sin(\pi t)$. The learning rate is also same as before. The result also shows the robot manipulator follows well the desired force. This experimental result also says that the neural network controller can adapt in almost 5 seconds. Of course, this adapting time depends on the learning rate of the neural network. If the learning rate is smaller, it takes more time to adapt.

The overshooting and oscillation of the force response can be seen in the all results. These are caused by not well learned system dynamics since it takes time to learn.

Conclusion

The application of neural networks to robot control is proposed and experimentally tested. The experimental results show that the neural network controller proposed in this paper is able to obtain the desired force in Cartesian coordinate. This ability is not only for fixed desired force, but also for a time varying desired force. Therefore we can apply this controller to obtain desired force response with an unknown environment. However, dynamics of the system is not learned immediately and causes overshooting of the force response. This overshooting, which might give damage to the environment, limits applications of this controller. This paper describes only the effectiveness and the possibility of application of neural networks to robot control to obtain desired force in Cartesian coordinate. The neural network controller proposed in this paper is not the best tuned controller. Further study is necessary for the best learning rate, how many hidden layers are required and how many neurons are required to perform better control. It is also important to study the best architecture of neural networks and the stability of the controller, since there are a lot of kinds of neural networks.

References

1. W. S. McCulloch, W. Pitts, "A Logical Calculus of the Idea Immanent in Nervous Activity," *Bull. Math. Biophysics*, 5, pp.115-133, 1943.
2. B. Widrow, "Generalization and information storage in network of adaline 'neurons'," *Self-Organizing Systems - 1962*, pp435-462, Spartan Books, 1962.
3. M. Kawato, K. Furukawa, R. Suzuki, "A Hierarchical Neural-Network Model for Control and Learning of Voluntary Movement," *Biological Cybernetics*, 57, pp169 - 185, 1987.
4. M. Kawato, Y. Uno, M. Isobe, R. Suzuki, "Hierarchical Neural Network Model for Voluntary Movement With Application to Robotics," *IEEE Contr. Syst. Mag.*, Vol.8, pp8-16, 1988.
5. K. Narendra, K. Parthasarathy, "Identification and Control of Dynamical Systems Using Neural Networks," *IEEE Trans. on Neural Networks*, Vol.1, No.1, pp4-27, 1990.
6. T. Yamada, T. Yabuta, "Direct Controller using Neural Network," *Trans. SICE'89*, pp391-392, 1989. (in Japanese)
7. T. Yabuta, T. Yamada, "Possibility of Neural Networks Controller for Robot Manipulators," *Proc. of IEEE Int. Conf. on Robotics and Automation*, pp1686-1691, 1990.
8. T. Yamada, T. Yabuta, "Some Remarks on Characteristics of Direct Neuro-Controller with Regard to Adaptive Control," *Trans. SICE*, Vol.27, No.7, 1991, pp784-791. (in Japanese)
9. T. Fukuda, T. Kurihara, M. Tokita, T. Mitsuoka, "Position and Force Hybrid Control of Robotic Manipulator by Neural Networks (1st Report, Application of Neural Servo Controller to Stabbing Controll)," *Trans. JSME*, 56-527, C, pp210-216, 1990. (in Japanese)
10. T. Fukuda, T. Shibata, M. Tokita, T. Mitsuoka, "Position and Force Hybrid Control of Robotic Manipulator by Neural Networks (2nd Report, Control of 2DOF Manipulators by a Neural network having Time Delay Elements)," *Trans. JSME*, 57-535, C, pp866-873, 1991. (in Japanese)
11. T. Fukada, T. Shibata, "Neural Network Applications for Robotic Motion Control," *Journal of Robotics and Mechatronics*, Vol.2, No3, pp21-25, 1991.

12. M. H. Raibert, J. J. Craig ,"Hybrid Position/Force Control of Manipulators," *Trans. ASME Journal of Dynamic Systems*, Measurement, and control, Vol.102, pp126-133, 1981.

13. K. Nakano, *An Introduction to Neurocomputing*, 1990. (in Japanese)

905203

Workshop Summary

COL Conrad Mialkowski,
Acting Director-General Research and Development Operations
CRAD[1] Headquarters, Constitution Building
305 Rideau Street
National Defence Headquarters, Ottawa, Ontario, K1A 0K2.

#149759

Abstract

To close the Workshop, COL Conrad Mialkowski, acting for Mr. Ken Peebles, Chief Research and Development, offered his impressions of the progress made during the week. He reviewed the highlights of the workshop, offered constructive criticism of the event and offered thanks to the organizers and workers.

Introduction

Well guys and gals, you done good! We've had a good (almost) three days of excellent interchange. I overheard lots of comments of the kind: "When I get back home, I'm going to try XXX" And that, in part, is what this is all about. Workshops are a stimulant to the imagination, and provide a motivation to use that person's ideas to get to the ultimate solution to all the world's problems before the next meeting. These workshops provide new sparks and also fan the flames to keep the creative process alive.

As you may not be aware Mr. Ken Peebles—CRAD—is delighted not to be here this morning. He is celebrating the joys of Paternity Leave and the birth of his new daughter, Emily. He has therefore asked me to fill in for him and provide this retrospective of the workshop.

In my comments this morning I'm going to say what I think was good about this event, but I'm also going to be a little critical—I hope in a constructive way.

Review

Major-General Fischer, in his opening address, described the context in which we find ourselves today. One aspect of that was a significant emphasis on "low cost". He said we must retain technological superiority at minimal cost. Keep it "high tech" but keep the cost down. This theme was reflected in a number of papers. We heard about the innovative usage of mass-produced commercial distance-measuring sensors with minor modifications to solve needs for space craft. We heard in the Power Sources session, and elsewhere, about the need for efficiency, the fact that roughly one-quarter of the life-cycle cost of a system is energy and therefore we must minimize its consumption, and so on. Clearly, keeping costs down is a recognized requirement for future military systems, and that includes robotic systems and capitalizing on knowledge-based systems.

The General also challenged us to choose our problems carefully. He told us not to waste time and effort on the impossible, but at the same time, to not to let the seemingly impossible hold us back. And there were some good examples of that here. Some of the most impressive demonstrated accomplishments of AI[2] are in solving large and formerly highly labour-intensive, routine management problems such as logistic re-supply and for aircraft loading. In tasks like these it has become evident that the products of scientists and engineers like us *are* going to reduce the need for manpower as well as lower the cost of operations.

Bob Fischer challenged us to let robots break out of the structured factory environment. And we did indeed hear about increasingly versatile robotic devices for space use and about new vehicles such as ANT[3] that effectively make some obstacles disappear. We *are* guiding robots out of the structured domain.

The General made reference to the need for international collaboration, and the value of that too, was highlighted in our Overview of Robotics.

Our session on Fuzzy Logic was unfortunately canceled, but this by no means signified a lack of interest or activity in this area. In my travels amongst the sessions, I heard many references to the applications of Fuzzy Logic. Had we put all those references together we would have had more than one session on the topic.

The TeleRobotics/Navigation session highlighted our need to understand the dynamics of operators and machines in the same system. Talk of the operator-machine interface is not enough; we need to consider to two in the same system.

---

[1] CRAD = Chief, Research and Development

[2] AI = Artificial Intelligence

[3] ANT = Articulated Navigation Testbed.

Our five invited speakers all provided us with much food for thought. General Fischer provided us with the challenge of the future in the military context. Jack Smith provided insight on how our world is going to change as a direct result of our own and other technological efforts. Jim Bowen provided new insight on how complementary neural networks and rule-based systems can be combined to solve problems better, or to solve problems that were previously intractable. Riccardo Patrella impressed upon us all the need to consider the global and ethical consequences of the development and management of new technology. And just this morning Martin Brooks provided us with a sense of accomplishment in the activities of Canadian robotics companies. I commend all of these speakers for widening our horizons and for reminding us of the need to consider factors outside of our immediate scientific interests.

As a participant in one of the Panel discussions I thought you all did a wonderful job asking us tough questions, and I was frankly amazed at the ability of the *other* panelists to answer them!

Towards Next Time

There are perhaps two areas where we might have done better.

I am disappointed not to have seen greater numbers of my colleagues in uniform here this week. We could not have located this workshop much closer to their normal place of work. And certainly we invited them to come and listen. But let's not blame them. What might we have done to win their interest? I think we should have been more aggressive in soliciting their input. We might have brought them in here personally and implored them to tell us more about their problems today. And perhaps we could have invited more of them to chair our technical sessions.

On the second matter I might be premature in this observation, but I have to admit that I didn't see the dynamic interchange between the Knowledge-Based Systems and Robotics communities that I had hoped to see. Now perhaps that will develop as a result of this joint workshop, but I think we are going to have to be more pro-active on this matter: we'll need to plan and structure the desired interaction.

This is the first time we have held this combined workshop, and on the whole I'm really pleased with the result. It has been an invigorating experience for me. We've come a long way, our momentum is up, and we're going to create those solutions two and three at a time! Keep up the good work.

Thanks

Before I finish, I'd like to add my thanks and indicate our appreciation to those who contributed to the success of this workshop. You may not know this, but it took a concerted personal effort on the part of CRAD himself—Mr. Ken Peebles—to make this event happen. In these times of resource reductions and political sensitivities, it required his active support to convince some senior people that this was the right thing to do. As Ken said in his opening, we thank all those who contributed through presenting papers and/or participating in the discussions.

The technical program was constructed through the hard work and leadership of Prof. Farooq of RMC[4], Dr. Pierre Roberge, also of RMC, and Dr. Julius Grodski of DCIEM[5]. These men, the members of the Program Committee, and several additional reviewers—I don't know who you all were at this point, but we'll try to see that you are acknowledged in the Proceedings—all these people put in a lot of hours to make this gathering a truly productive and significant event.

There are several people in CRAD Headquarters who made sure you were registered and that the facilities were in place for your use. The registration team of Bob Elliott, Dorothy Hill, and Linda Baggs were most ably assisted by Ms Margaret Coll and Jennifer Weichel. Robert Pilon, Wally Blair and Maj Alan St-Onge looked after the meeting rooms, the lunches and the coffee, as well as supervising a staff of a number of volunteer projectionists. Sgt Louise Morin helped out in a number of ways including as a projectionist. Ms Cony Kluver is cataloguing your papers for the Proceedings; the list goes on. This is sounding just like the Academy Awards!

Bob Dobson has been a driving force behind these meetings for a long time. Besides helping to get this workshop on the rails initially, he was responsible for organizing the Exposition, for coordinating these events interdepartmentally, and for arranging the invited speakers. I'm sure you will agree that the presence of the Exposition added significantly to the "ambiance".

And finally, Dr. Harold Merklinger whose personal commitment and determination were pivotal to the success of these three days—very well done Harold!

Please join me in giving all these folks a round of applause!

---

[4] RMC = Royal Military College

[5] DCIEM = Defence and Civil Institute of Environmental Medicine.

Author	Page#
Abu-Hakima, S.	557
Adamovits, Peter	245
Adams, W. A.	189
Akhras, G.	367
Amyot, J. R.	515
Anantaraman, A. V.	189
Ananthanarayanan, S.	93
Andrews, W. S.	85
Archibald, Colin	309
Ball, Roy	47, 383
Basran, Jagdeep S.	293
Bassias, Constantinos	587
Baudoin, Yvan	419
Bayoumi, M. M.	641
Bélanger, Micheline	153, 159
Bhavsar, Virendra C.	137
Blais, F.	633
Blezius, Jeffrey	63
Bock, O.	101
Boev, W.	515
Boily, Jocelyn	345
Boucher, J.F.	329
Boulanger, Pierre	731
Bourassa, Gregory	601
Bowen James E.	375
Bowen, B. Archie	71, 375
Brooks, Martin	655
Brosinsky, Chris	649
Brown, Richard G.	159
Bryden, Karen	471
Burtnyk, Nestor	253
Caldwell, D.E.	509
Carling, Robert L.	145
Carrel, Aubert	461
Carruthers, Al	283
Chaib-draa, B.	625
Chalmers, Bruce A	677
Chrystall, Keith	63, 315
Cohen, P.	173
Cox, David S.	85
Croteau, Danie	389
Cunningham, R. G.	197
Cyre, Walling R	323
Cyril, Xavie	573
Dagnino, Aldo	315
Dai, Heping	739
Damaren, Chris	547
Das, Yoga	283
Dawes, Nick	77
Dawson, C. P.	123
de Moulins, J.-L.	461
Dégé, Michel	461
Dent, Ann	657
Desbiens, Jocelyn	485, 593
Desrochers, Alain	55, 389
Dimopoulos, N.J.	5
Drascic, David	261, 269
D'Eleuterio, G.M.T.	101
Eirich, Ron	283
Elgazzar, S.	633
ElMaraghy, Hoda A.	107
Falcou, Gérard	461
Farooq, M.	541, 641
Feighan, Patrick	315
Fenrick, Dennis R.	649
Ferguson, James	541 (Farooq)
Ferland, Martin	581
Fischer, MGen R. N.	1
Fisher, G. F.	493
Folinsbee, Terry	383
Foo, H. C.	367
Fox, Mark S.	13
Freedman, Paul	277
Fugere, Jean	225
Gagné, Denis	485, 593
Garant, Alain	485
Gaskill, H. S.	123
Gauvin, Marlène	237
Gely, Paul	709
Gill, S.	173
Goldenberg, A. A.	93
Goldfarb, Lev	137
Graham, Barry	345
Graham, Darin P.	253
Green, D. N.	183
Green, D. A.	633
Grodski, Julius J.	101, 261, 269, 429, 435, 447, 701
Hall, Bart	701
Hanna, Doug M.	649
Hardy, Gordon	565
Harris, Gail E.	501, 685
Hay, Robert	329
Hayashi, Robert	657
Helie, J.	329
Hemami, Ahmad	693
Henneker, W.H.	611
Hensler, John	211
Hinchey, Michael	167
Huculak, Michael	573
Hurteau, R.	173
Jenkins, L.	723

Author	Page#
[illegible]	[illegible]
Johnson, Ivan	13
Jones, Stephen	587
Jones, Major S.	723
Kamat, Vithal N.	137
Kennedy, J. L.	329
Kiguchi, Kazuo	747
Kim, B.	183
Korba, Larry	39
Korelsky, T.	509
Kramer, Al	283
Krouglicof, N.	55, 389
Labbé, Gaétan	329
Labbé, Marcel	21
Laidley, R. G.	205
Lamontagne, Luc	477, 625
Laplante, Martin	501, 685
Lecavalier, M.	633
Lelarge, Annie	461
Levesque, P.	625
Lewis, B.J.	85
Li, Kin F.	5
Liang, Feng	107
Lind, Peter C.	715
Lipitkas, J.	101
Liu, Jianli	71, 375
Lizotte, Michel	237
Lortie, Martin	709
Lu, Z.	93
Marrocco, Bernard	345
Malowany, A	351
Manz, A	197
Mason, R. D.	197
Matthews, J. R.	329
McCormac, Alan W.	649
McMath, W. S.	39, 411
Megley, Rick	573
Mialkowski, Col C.	755
Mikolajuk, Zbigniew	529
Milgram, Paul	261
Miller, S. J.	397, 555
Murgulescu, M.C.	301
Mylopoulos, J.	93
Nault, Georges	485
Necsulescu, Dan S.	183, 747
Nelson, Terrence J.	179, 405
Neville, Stephen	5
O'Shea, Jules	581
Pagurek, Bernard	77
Paquet, E.	625
Paquin, Normand	63
Peebles, K. A.	[illegible]
Pelkowitz, Lionel	47
Penzes, Steve	47
Petriu, Emil M.	293, 309, 411
Phan, C. S.	611
Phelps, Russel M.	435
Piedboeuf, J.-C.	173, 397, 555
Poehlman, W. F. S.	715
Poulin, Jean	709
Pouplard, Jean-Pierre	461
Preston, Jon M.	115
Pryor, Tim	383
Rabemanantsoa, M.	581
Rastogi, Anu	261
Richard, Denis	55
Richardson, Capt J.	21
Roberge, P. R.	29, 205, 493, 617
Roy, Dominic	237
Saks, Victor	13
Salvador, J. P.	205
Sandrasegaran, K.	351
Sasiadek, J. Z.	183
Sassine, Raymond	587
Selkirk, C. G.	493
Seifu, Solomon	253
Sfeir, Daphne	63
Sharf, Inna	547
Shupe, Landy	115
Smith, J. E	131
Solem, Eric	523
Suen, Ed	657
Swinson, M.L. LtC	441
Taylor, Ivan	219
Thomas, Grant	537
Thompson, J. Walter	739
Tinney, LCdr Mark	211
Toulouse, Michel	593
Tousignant, C.	587
Trethewey, K. R.	29
Trif, N.	411
Viens, Capt François	77
Watkins, Andrew	5
Wattie, Dwaine	471
Weisman, Maj R.	225
Wojcik, Peter	63
Woods, Steven	665
Yamakawa, D.	5
Yaniv, Elan	323
Yeung, S. K.	39, 411, 493
Zhao, Hong	219
Zhao, C. S.	641